

**BIOINSPIRED ALGORITHM FOR IDENTIFYING OVERLAPPING  
CLUSTERS IN PROTEIN-PROTEIN INTERACTION NETWORKS**

BY  
**AHMED ABDULGLIL DAEL NAEF**

A Thesis Presented to the  
DEANSHIP OF GRADUATE STUDIES  
**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**  
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the  
Requirements for the Degree of

**MASTER OF SCIENCE**

In

**COMPUTER SCIENCE**

**MAY, 2014**

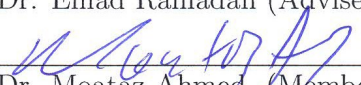
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS  
DHAHRAN 31261, SAUDI ARABIA

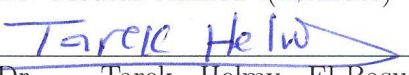
DEANSHIP OF GRADUATE STUDIES


This thesis, written by **AHMED ABDULGLIL DAEL NAEF** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER SCIENCE**.

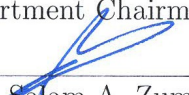
Thesis Committee

  
Dr. Emad Ramadan (Adviser)

  
Dr. Moataz Ahmed (Member)

  
Dr. Tarek Helmy El-Basuny  
(Member)

  
Dr. Ahmed F. Adel  
Department Chairman

  
Prof. Salam A. Zummo  
Dean of Graduate Studies

27/5/14  
Date



©Ahmed Naef  
2014

*This Thesis is dedicated to  
My parent, wife, cute little daughter, brothers and sisters for their  
affection, love and encouragement.*

# ACKNOWLEDGMENTS

*First and foremost, I would like to sincerely thank almighty Allah for all his grants that he bestowed on me. Secondly, I have to thank the last prophet Mohammed (peace and blessing be upon him) who taught us the things we ought to do.*

*My deep appreciation and heartfelt gratitude to my thesis adviser Dr. Emad Ramadan and my thesis committee member Dr. Moataz Ahmed for their excellent guidance, caring, patience, constant endeavour, and providing me with an excellent atmosphere for doing research. I would also like to thank my thesis committee Dr. Tarek Helmy El-Basuny for his encouragement and insightful comments.*

*I also wish to express my sincere thanks to the Department of Information and Computer Science for their support and assistance by providing all the available facilities, especially the head of department Dr. Ahmed Adel.*

*I owe thanks to my colleagues and my friends for their help, motivation and pivotal support.*

*Finally, and most importantly, my heartfelt gratitude is given to my beloved father, mother, wife, my cute little daughter, brothers and sisters for their unwavering love, support, patience, encouragement, and faith in me.*

# TABLE OF CONTENTS

ACKNOWLEDGMENT	iii
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF ABBREVIATIONS	x
ABSTRACT (ENGLISH)	xi
ABSTRACT (ARABIC)	xiii
CHAPTER 1 INTRODUCTION	1
1.1 General . . . . .	1
1.2 Problem Statement . . . . .	5
1.3 Thesis Objectives . . . . .	5
1.4 Thesis contributions . . . . .	6
1.5 Thesis Methodology . . . . .	6
1.6 Thesis Outline . . . . .	7
CHAPTER 2 BACKGROUND AND OVERVIEW	9
2.1 Biological background . . . . .	9
2.1.1 Cells biology . . . . .	9
2.1.2 Genome . . . . .	10
2.1.3 Proteins . . . . .	12

2.2	Genetic Algorithm . . . . .	15
2.3	Spectral Clustering . . . . .	16
<b>CHAPTER 3 LITERATURE REVIEW</b>		<b>18</b>
3.1	Affinity Propagation-Based Methods . . . . .	18
3.2	Density-Based Methods . . . . .	19
3.3	Model-Based Methods . . . . .	23
3.4	Random Walk-based Methods . . . . .	24
3.5	Genetic Algorithm-Based Methods . . . . .	25
<b>CHAPTER 4 A NEW GA BASED CLUSTERING APPROACH</b>		<b>27</b>
4.1	Research Problem . . . . .	27
4.2	Research Approach . . . . .	29
4.2.1	Introduction . . . . .	29
4.2.2	Representation and Initialization . . . . .	30
4.2.3	Objective Function . . . . .	34
4.2.4	Genetic Operators . . . . .	39
<b>CHAPTER 5 EXPERIMENTS AND RESULTS</b>		<b>43</b>
5.1	GA parameters setup and optimization . . . . .	43
5.2	Results Analysis . . . . .	44
5.2.1	Data Set . . . . .	44
5.2.2	Cluster validation based on known complexes . . . . .	45
5.2.3	Cluster validation based on functional homogeneity . . . . .	63
<b>CHAPTER 6 CYTOSCAPE PLUGIN (BIOCM)</b>		<b>69</b>
6.1	Installation . . . . .	69
6.2	Running BioCM . . . . .	73
<b>CHAPTER 7 CONCLUSION AND FUTURE WORK</b>		<b>77</b>
<b>REFERENCES</b>		<b>79</b>





# LIST OF TABLES

2.1	The 20 amino acids (three-letter amino acid code) corresponding to each codon. . . . .	14
4.1	The objective functions used in previous published works. . . . .	34
4.2	An illustrative example of the four objective functions used. . . . .	39
5.1	GA parameters setup using four different objective functions to compute the fitness values of the population. . . . .	44
5.2	Comparison of Clustering Algorithms on the Yeast Collins Network.	46
5.3	The average, standard deviation, max and min of the recall, precision and f-measure validation scores used to validate the resulted clusters of 5 runs [the 1st population of GA is generated randomly].	54
5.4	The average, standard deviation, max and min of the recall, precision and f-measure validation scores used to validate the resulted clusters of 5 runs [the 1st population of GA is generated using the clusters resulting from spectral clustering algorithm]. . . . .	55
5.5	A few of the clusters in Collins network with the lowest $p$ -values with GO components. . . . .	64

# LIST OF FIGURES

1.1	A graph modeling protein-protein interaction network . . . . .	3
2.1	Anatomy of the Animal Cell [1]. . . . .	10
2.2	DNA helix. . . . .	11
2.3	3D structure of a protein. . . . .	13
2.4	Central dogma. . . . .	13
4.1	Chromosome representation [for our clustering approach]. . . . .	31
4.2	Population initialization method [using the random initialization method and spectral clustering method]. . . . .	33
4.3	A Cluster representing ratio cut limitation. . . . .	36
4.4	A Cluster representing normalized cut limitation. . . . .	36
4.5	A Cluster representing max-min cut limitation. . . . .	37
4.6	Mutation operation. (a) shows the selected node of the cluster $c_i$ . (b) shows the cluster $c_j$ after the mutation operator. . . . .	41
4.7	Mutation operation. (a) shows the selected node of the cluster $c_i$ . Figure (b) illustrates the cluster $c_i$ after adding the selected node's neighbors from the graph $G$ . . . . .	42
5.1	Comparative results of the considered clustering approaches using Precision measure. . . . .	48
5.2	Comparative results of the considered clustering approaches using Recall measure. . . . .	49

5.3	Comparative results of the considered clustering approaches using f-measure. . . . .	50
5.4	The percentage of discarded proteins in the Collins network. . . .	51
5.5	Best fitness value of four objective functions for a particular run. .	58
5.6	Average of fitness values of four objective functions for a particular run. . . . .	59
5.7	Standard deviation of fitness values of four objective functions for a particular run. . . . .	60
5.8	Average of the best fitness values of four objective functions over 10 runs. . . . .	61
5.9	Standard deviation of the best fitness values of four objective functions over 10 runs. . . . .	62
5.10	Clusters size distribution. . . . .	66
5.11	Clusters density distribution. . . . .	67
5.12	Clusters degree distribution. . . . .	68
6.1	Plugin manager in Cytoscape. . . . .	70
6.2	The installation of BioCM plugin. . . . .	71
6.3	The installed BioCM plugin. . . . .	72
6.4	The input of BioCM plugin. . . . .	73
6.5	Running BioCM plugin. . . . .	74
6.6	SnapShot of the output of BioCM plugin. . . . .	75
6.7	SnapShot of a visualization of the predicted clusters [using Matlab and Mathematica functions]. . . . .	76

## LIST OF ABBREVIATIONS

DC: Density Cut Objective Function.

MC: Max-Min Cut Objective Function.

NC: Normalized Cut Objective Function.

RC: Ratio Cut Objective Function.

RB: (Random-Based), the first population of genetic algorithm Based on Random method.

SB: (Spectral Clustering-Based), the first population of genetic algorithm Based on Spectral Clustering Algorithm.

# THESIS ABSTRACT

**NAME:** Ahmed Abdulglil Dael Naef

**TITLE OF STUDY:** Bioinspired Algorithm for Identifying Overlapping Clusters in Protein-Protein Interaction Networks

**MAJOR FIELD:** Computer Science

**DATE OF DEGREE:** May, 2014

*Recently, biological networks have attracted a lot of researcher efforts as they are very essential in increasing our knowledge of living systems at the cellular level. Consequently, several methods have been developed to study and analyze the topological features of such networks.*

*In this work, we focus on particular biological networks, called protein-protein interaction networks (PPI) which obtained by using recent technologies such as yeast-two hybrid and mass spectrometry as well as several computational models. We develop algorithms for studying these networks. We aim to assist biologists to draw a conclusion about the general principles that control all the biological processes for producing a correctly functioning organism. The applications of the existing clustering methods applied on these networks would not gain good*

*findings due to the scale-free structure, small-world, disassortivity and multifunctionality characteristics of PPI networks. We consider a genetic algorithm technique to develop a computational model for identifying functional modules in PPI network. We assess the quality of our findings whether they have any biological meaning by using gene ontology terms. Furthermore, we compare and validate the performance of our clustering approach with three competing clustering methods: MCL, MCODE and ClusterOne. Our analysis of the clusters identified demonstrates that our clustering approach: (a) can find several biologically significant protein complexes (group of proteins) compared to cellular component GO terms; (b) group higher percentage of proteins in the original network; and (c) is more effective than existing approaches (i.e., MCL, ClusterOne, and MCODE) when compared against two reference sets: MIPS and CYC2008.*

## ملخص الرسالة

الاسم الكامل: احمد عبدالجليل دائل نائف

عنوان الرسالة: تصميم خوارزمية لإيجاد الوحدات الوظيفية في شبكات التفاعل بين البروتينات

التخصص: علوم حاسب آلي

تاريخ الدرجة العلمية: مايو-2014

نظرا لأهميتها البالغة في فهم الأنظمة الحيوية على المستوى الخلوي؛ احتلت دراسة الشبكات البيولوجية وتحليلها — لا سيما في السنوات الآتية — عناية فريدة واهتماما متميزا من قبل العديد من الباحثين؛ الأمر الذي استدعى تطوير العديد من الخوارزميات لدراسة هذه الشبكات وتحليلها.

تسلط هذه الدراسة تركيزها على نوع واحد من الشبكات البيولوجية وهي: شبكة التفاعلات بين البروتينات والتي يمكن الحصول عليها من خلال استخدام بعض التقنيات مثل: Yeast-two hybrid و Mass spectrometry بالإضافة الى العديد من النماذج الحسابية. فاعتمادا على الخوارزميات الجينية (Genetic Algorithm)، تقترح هذه الدراسة خوارزمية لدراسة شبكة التفاعلات بين البروتينات من خلال تصنيف البروتينات الى مجموعات تسمى (clusters)؛ حيث إن البروتينات في كل مجموعة لديها وظيفة بيولوجية محددة. كما تجدر الإشارة إلى أنه يوجد العديد من العيوب في التطبيقات المتوفرة والمعتمدة على الخوارزميات الحالية لتصنيف البروتينات؛ وذلك لأنها لم تأخذ بعين الاعتبار بعض خصائص هذه الشبكات مثل: scale-free structure, multifunctionality and disassortivity, small-world. ومن هنا يأتي هدف هذه الدراسة لمساعدة متخصصي الأحياء لفهم المبادئ العامة التي تتحكم في كل العمليات البيولوجية.

لقد قمنا في هذه الدراسة بتقييم نتائج الخوارزمية المقترحة عما إذا كانت تحتوي على أي أهمية بيولوجية عن طريق مقارنتها مع gene ontology terms، ثم مقارنة أداء الخوارزمية المقترحة مع خوارزميات أخرى: MCL, MCODE and ClusterOne. وبناءً على النتائج التي حصلنا عليها عند استخدام الطريقة المقترحة؛ يمكننا أن نقول: إن الطريقة المقترحة قادرة على الآتي:

(أ) إيجاد clusters ذات أهمية بيولوجية. (ب) تصنيف نسبة كبيرة من البروتينات الموجودة في شبكة التفاعلات بين البروتينات. (ج) كما تتسم هذه الطريقة بفاعلية أكثر من الخوارزميات الحالية ( MCL MCODE and ) . (ClusterOne



# CHAPTER 1

## INTRODUCTION

### 1.1 General

Proteins carry out all essential biological functions in all living organisms [2]. Studying the proteins as well as their interactions is very vital in order to understand how the proteins achieve their functions within a cell [3]. In the last few decades, high-throughput experimental methods such as: yeast-two hybrid [4] and mass spectrometry [5] have been used for discovering the pairwise protein interactions. These techniques and other inexpensive tools (computational models) have amassed a huge amount of data of protein-protein interaction networks. This work contributes only as a part of wider studies done to explore and analyze the proteome -all proteins that make up an organism.

In general, proteomic network data set is modeled as a graph  $G = (V, E)$ , as illustrated in Figure 1.1 where nodes  $V$  represent proteins and edges  $E$  represent interactions between the proteins. Several research studies have been

done to analyze the proteomic networks [3][6]. Such studies have uncovered several significant topological characteristics of the protein-protein interaction (PPI) networks, including *multifunctionality* a protein can be included in various biological processes, *small world property* which defined as follows: the distance  $l$  between two nodes is proportional to the logarithm of the network size  $N$ , *power-law degree distribution* which defined as follows: the probability  $p(k)$  that a node has  $k$  links to other nodes is  $p(k) \sim k^{-\gamma}$ , where  $\gamma$  is the degree exponent [7]. Such networks that have a power-law degree distribution are called *scale-free networks* in which a few proteins have lots of interactions and a lot of proteins have few interactions to other proteins and *disassortativity* in which proteins having a lot of interactions are not directly connected to each other.

According to the analysis performed on a number of published proteins interactions, the findings have shown that proteins of known functions tend to group together [8]. Thus, understanding the inner workings of the cells more clearly demands identifying protein clusters within a cell's biological network. Hence, developing effective methods for revealing the modular structure (protein clusters) in a graph modeling the PPI networks has become a major challenge in computational system biology.

In recent years, many clustering algorithms, depending on different approaches and ideas, have been developed for revealing protein complexes in the PPI networks. These algorithms can be classified into two categories: exclusive clustering algorithms and overlapping clustering algorithms. The algorithms (e.g.

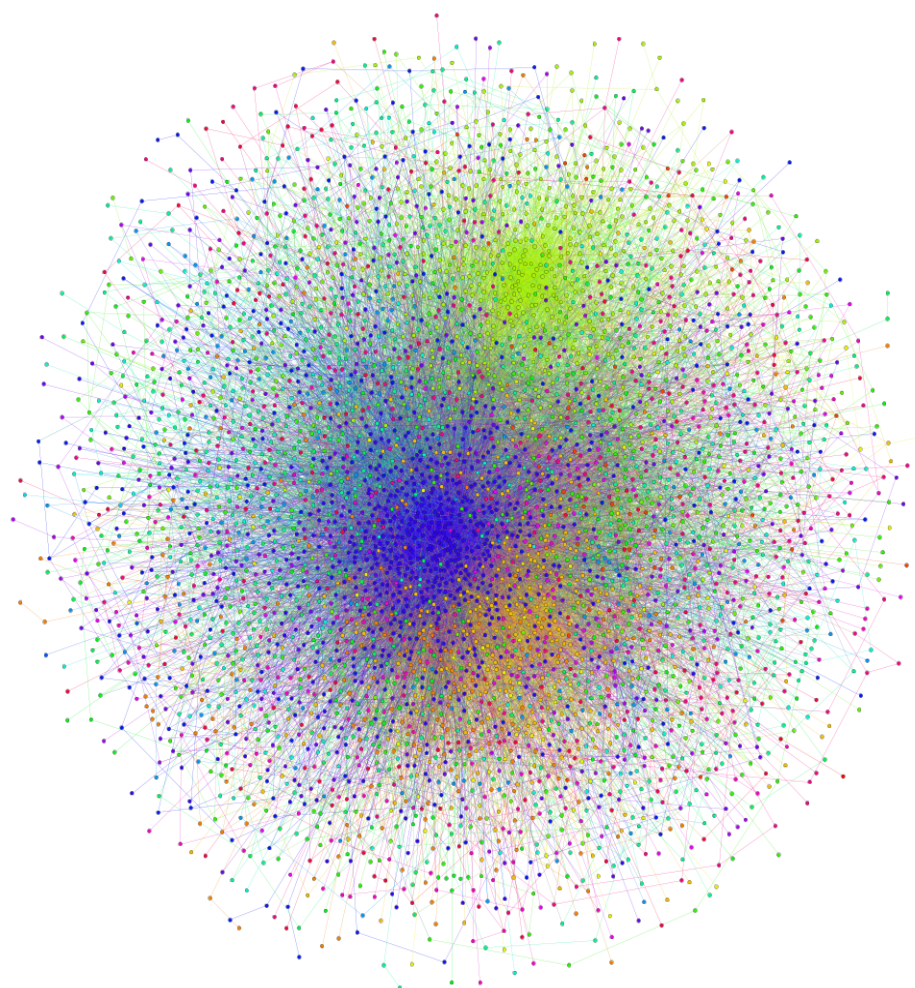


Figure 1.1: A graph modeling protein-protein interaction network

MCL [9], SPICi [10]) have the ability to discover exclusive complexes in which each protein must belong to at most one cluster while in biology one protein may be involved in many complexes simultaneously. (2) The other algorithms (e.g. MCODE [11], ClusterOne [12], CFinder [13] and OCG [14]) can discover overlapping clusters in which there are some common proteins in the identified clusters. In general, it has been observed that several clustering methods start looking for cliques, fully completed subgraphs, or densely connected subgraphs in the PPI networks in order to identify the overlapping or non-overlapping protein complexes. Here, we list some limitations of the considered clustering methods, regarding MCODE [11] and CFinder [13]: the sparsely interconnected clusters are neglected, the percentage of covered proteins is low, and their results either a small number of large clusters or a large number of small cliques. Although MCL [9] algorithm is more robust and scalable, it does not support finding the overlapping clusters. On the other hand, another algorithm [15] relies on messages passing between nodes which determine whether a pair of nodes may belong to the same cluster. The main limitation of this method is determining the best value of the parameter exemplar that gives an optimal clustering solution. Other algorithm (e.g. RSGNM [16] and RSRGM [17]) based on multiplicative updating rule [18] in order to optimize the protein-cluster membership which generated by another proposed method or generated randomly. Another clustering method (e.g. PRO-COMOSS [19]) uses a genetic algorithm for finding overlapping clusters and based on semantic similarity of gene ontology. The main drawback of this approach is

that the predicted clusters cover a small percentage of the PPI network.

In this work, we consider clustering such networks into complexes (groups of proteins) that share a common biological activity using the concept of Genetic Algorithm (GA) approach that take into account the topological characteristics of the proteomic networks. We give a basic overview of GA in section 2.2

## 1.2 Problem Statement

Cellular processes are achieved by multi-protein complexes/functional modules (communities). Several studies have shown that clustering PPI is an effective way for finding protein complexes. However, revealing the modular structure of such networks remains a major challenge in computational system biology.

**Research Question:** Can we discover the presence of communities in a network and identify the members of the communities?

## 1.3 Thesis Objectives

The goal of this study is to design algorithms for studying PPI networks to discover biologically significant clusters. Specific goals for this work are as follows:

- Detect the presence of communities in PPI networks (functional modules/protein complexes) and find the members of these communities.
- Help biologists to find the general principles that govern the organization of

protein-protein interaction networks.

## 1.4 Thesis contributions

In particular, the contributions of this study are:

1. Overlapping clustering for biological networks: We introduce a clustering approach which is effective for clustering networks with the following characteristics: scale-free structure, small-world, disassortativity and multifunctionality. Furthermore this clustering approach identifies clusters with varying properties: cohesive clusters (cliques or near-cliques), and non-cohesive clusters . This approach also has high coverage ratio.
2. Predict the cellular function of uncharacterized proteins.
3. Validate discovered protein clusters using two reference sets (CYC2008 and MIPS) and Gene Ontology terms.

## 1.5 Thesis Methodology

Here, we state the main tasks that have been done in order to achieve the stated objectives:

1. Literature Review:

We have conducted a critical survey of clustering approaches for identifying overlapping clusters in protein-protein interaction networks.

2. Collecting Materials:

We have collected and analyzed the biological networks data that are used in this study as well as the reference sets that are used for validation.

3. Developing GA-based clustering approach:

We have designed a clustering approach for identifying overlapping clusters in protein-protein interaction network using genetic algorithm method.

4. Developing a software tool:

We have developed a cytoscape plugin that packages our clustering algorithms.

5. Performance analysis:

We have compared the performance of our clustering approach with three competing clustering approaches. Furthermore, we have evaluated the quality of the resulted clusters compared with two reference sets and cellular component terms from gene ontology.

## 1.6 Thesis Outline

This thesis is organized as follows. Chapter 2 provides some biological concepts and an introduction to genetic algorithm and spectral clustering. A summary of the literature surveyed so far is provided in Chapter 3. Chapter 4 presents the developed clustering approach to identify protein complexes in protein-protein interaction networks. Chapter 5 addresses our experiments

and the results obtained by applying our clustering approach in order to identify protein complexes. Chapter 6 presents a cytoscape plugin that packages our clustering algorithm. Finally, Chapter 7 presents a general conclusion and suggests some future work.



# **CHAPTER 2**

## **BACKGROUND AND OVERVIEW**

This chapter presents some biological concepts and an introduction to genetic algorithm and spectral clustering.

### **2.1 Biological background**

Here we give a brief concept about molecular biology which is very important for understanding this thesis.

#### **2.1.1 Cells biology**

Cells are the fundamental unit of life. Every living organism - from the smallest bacterium to the largest mammal is made of one or more cells [2]. Cells, as shown in Figure 2.1 are enclosed by a plasma membrane, which separates the interior

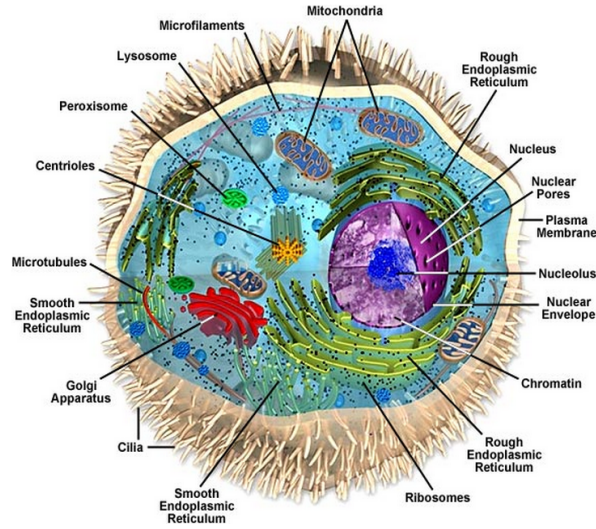


Figure 2.1: Anatomy of the Animal Cell [1].

contents of all cells in order to protect the cell from the surrounding environment and allow the materials to enter and leave the cell. Each cell contains a variety of components called organelles each with a specific function. The most important part in the cell is the nucleus which existed just in the eukaryotic cells. It is considered as the cell's instructions center that regulates all cell activities including sending instructions to the cell to grow, divide or die as well as regulating the gene expression process - *the process by which information from a gene is used to synthesize a functional gene product(Protein)*.

### 2.1.2 Genome

Deoxyribonucleic acid (DNA) is the hereditary material in almost all living organisms. It is a molecule which stores all genetic information needed to make and regulate all organisms. DNA is arranged in two long complementary strands that form a double helix as illustrated in Figure 2.2. The complete set of DNA

molecules is called *organism's genome*. DNA is encoded as a sequence of four chemical bases (nucleotide): adenine (A), thymine(T), guanine (G) and cytosine (C). Such bases pair up with each other following a set of rules: A pairs with T and C pairs with G, to form units called base pairs.



Figure 2.2: DNA helix.

### 2.1.3 Proteins

Proteins are produced using the information encoded in DNA sequence. The process of manufacturing proteins is called central dogma (gene expression) as shown in Figure 2.4. This process involves two main operations: First, RNA transcription in which enzymes called RNA polymerases read the information in a relevant region of DNA molecule - which is usually for a single protein - and transcribe it into a messenger ribonucleic acid (mRNA) chain. mRNA is encoded as a sequence of four chemical bases (nucleotide): adenine (A), thymine(T), guanine (G) and uracil (U). Second, protein translation in which every consecutive three bases in mRNA is translated into an amino acid according to standard genetic code illustrated in Table 2.1, which in turn a chain of amino acid make up a particular protein. The proteins are the essential working parts of organisms which playing the main functions in almost all processes of life [2]. Proteins achieve their biological functions within a cell by forming multi-protein functional modules (complexes), which are groups of proteins. Thus, knowing such complexes provides a greater understanding of cellular functions and organization. Such predictions can be done through high large-scale experiments or inexpensive computer modeling tools.

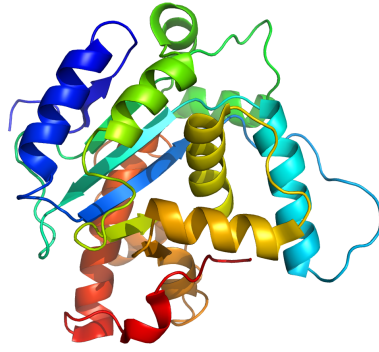


Figure 2.3: 3D structure of a protein.

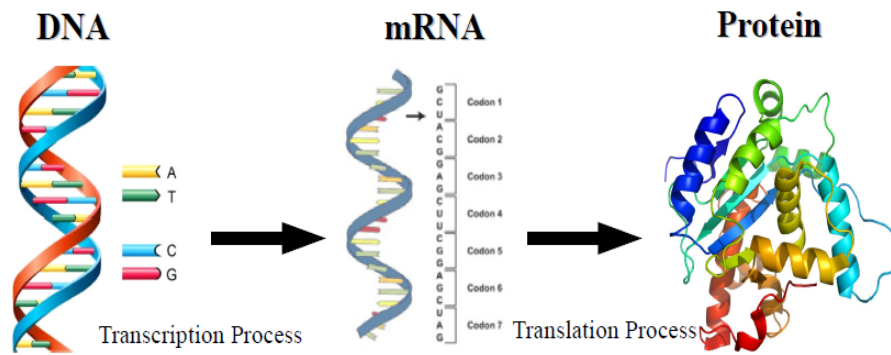


Figure 2.4: Central dogma.

Table 2.1: The 20 amino acids (three-letter amino acid code) corresponding to each codon.

		Second letter of the codon								
		U		C		A		G		
First letter of the codon	U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U
		UUC	Phe	UCC	Ser	UAC	Tyr	UGC	Cys	C
		UUA	Leu	UCA	Ser	UAA	Tyr	UGA	Stop	A
		UUG	Leu	UCG	Ser	UAG	Tyr	UGG	Trp	G
	C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg	U
		CUC	Leu	CCC	Pro	CAC	His	CGC	Arg	C
		CUA	Leu	CCA	Pro	CAA	Gln	CGA	Arg	A
		CUG	Leu	CCG	Pro	CAG	Gln	CGG	Arg	G
	A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U
		AUC	Ile	ACC	Thr	AAC	Asn	AGC	Ser	C
		AUA	Ile	ACA	Thr	AAA	Lys	AGA	Arg	A
		AUG	Met	ACG	Thr	AAG	Lys	AGG	Arg	G
	G	Ala	Val	GCU	Ser	GAU	Asp	GGU	Gly	U
		GUC	Val	GCC	Ala	GAC	Asp	GGC	Gly	C
		GUA	Val	GCA	Ala	GAA	Glu	GGA	Gly	A
		GUG	Val	GCG	Ala	GAG	Glu	GGG	Gly	G
		Third letter of the codon								

## 2.2 Genetic Algorithm

Genetic Algorithms (GAs) are a family of search and optimization methods which are based on Darwin's theory (evolutionary theory) "survival of the fittest" [20]. GA was first invented by John Holland in 1975. Following that, it has been improved by a number of researchers [21] [22]. It has proved to be a highly suitable and powerful method to progress toward an approximate solution which should be as close as possible to the optimal solution in search and optimization problems. It mimics the processes inspired from biological evolution such as inheritance, selection, mutation, and crossover. Initially, the genetic algorithm approach depends on a population (a set of individuals) where each individual represents a candidate solution for a given problem. Based on fitness functions, selection, crossover and mutation operators, the population is refined in each generation by selecting the fittest individuals and modifying them to generate a new population for the next generation. The fitness value of each individual indicates how well each individual is suited to be a solution. Consequently, GA reaches to a satisfactory individual as a solution to a problem. The common steps of GA are the following [23] :

1. Create the initial population of possible solutions (individuals).
2. Compute the fitness value of each individual.
3. Select all the individuals, that are used as parents to create the next generation, based on their fitness values and the selection method.
4. Make perturbation to each of these selected individuals using genetic op-

erators, e.g. crossover and mutation to create the offsprings of the next generation.

These steps, except population initialization step, are iterated until some stopping criteria are satisfied. Before GA can be used, there are four domain-dependent things to do: representing the chromosome of the problem, the blueprint of the possible solutions, which supposed to be very close to the original solution of the considered problem, defining the fitness function, selecting parent selection methods, and defining genetic operators.

## 2.3 Spectral Clustering

A clustering of a graph is a partitioning of the vertices into groups such that vertices in each group are similar to each other and dissimilar to vertices in other groups. In an exclusive clustering, each vertex belongs to at most one subset in the clustering, but in an overlapping clustering, a vertex could belong to more than one subset. Here, we present a brief introduction to the family of spectral clustering methods which have been applied widely over the last decades and several algorithms have been proposed along this line of study. There are some intuition behind the popularity of such spectral clustering approaches which can be summarized as follows: such approaches are very simple to implement as they based on standard algebra methods [24]. Furthermore, they have the ability to figure out problems in much complex shapes such as spiral, linear and nonlinear shapes as they are invariant to cluster shape, that is, they do not make



presumption to the clusters' shapes. Algorithm 1 illustrates the spectral clustering algorithm used in this study.

---

**Algorithm 1** Normalized Spectral Clustering.

---

- 1: Given an adjacency matrix  $A$ .
  - 2: Construct the degree matrix  $D$ , the degree for each vertex is computed by the number of adjacent vertices of that vertex  $d_i = \sum_{j=1}^n A_{ij}$ .
  - 3: Compute the symmetric Laplacian matrix  $L$ .
  - 4: Identify  $v_1, v_2, \dots, v_k$  the top  $k$  eigenvectors of  $L$ .
  - 5: Construct the matrix  $V \in \mathbf{R}^{n \times k}$  from  $v_1, v_2, \dots, v_k$ .
  - 6: Each row of  $V$  represents a vertex in  $\mathbf{R}^k$ , group these vertices into  $k$  clusters using any approach such as *k-means* algorithm
-

## CHAPTER 3

# LITERATURE REVIEW

Several studies have been done on the the problem of clustering PPI network to identify protein complexes. Although there are a wide variety of such methods in the literature, this review can be divided into groups including density-based clustering methods [14], [25], [11], [12], [26], [13], [27], message passing-based clustering method [15], random walk-based method [9] and genetic algorithm-based clustering method [19]. In this Chapter, we discuss and review in brief such clustering algorithms.

### 3.1 Affinity Propagation-Based Methods

Wang and Gao [15] proposed an algorithm called Overlapping Affinity Propagation (OAP) for identifying overlapping complexes in PPI network. This algorithm based on passing messages. The first message is responsibility message  $r(i, k)$  sent from a data point to a candidate exemplar which indicates how strongly the data point  $i$  prefers the exemplar  $k$ . The second message is availability message

$a(i, k)$  sent from an exemplar to a data point which indicates the probability of the node  $k$  to be available as an exemplar to the data point  $i$ . Given the adjacency matrix  $A_{N \times N}$ , where  $N$  is the number of proteins in the PPI network. OAP algorithm involves the following phases: (i) computing the similarity  $S_{N \times N}$  between each pair of vertices using Jaccard similarity measure, i.e.  $s(i, k)$  indicates how well the vertex with index  $k$  is suited to be the exemplar of the vertex  $i$  and the diagonal of the similarity matrix represents the prior exemplars, preferences; (ii) obtaining the exclusive clusters of the graph using AP algorithm; (iii) initializing the availability matrix  $A_{N \times N}$  to zero; (iii) continue updating the availability and responsibility matrices until reaching to a steady state; and (iv) the vertices that share the same exemplar are considered as a cluster as well as the proteins that have more than one exemplar are considered as candidate overlapping vertices as long as satisfying some conditions. The drawback of such a method is how to determine the number of the preferences.

## 3.2 Density-Based Methods

There are several clustering methods in the literature start looking for either cliques, fully completed subgraphs, or densely connected subgraphs in the PPI networks in order to identify the overlapping modules in the studied networks. In this section, we show some density-based methods.

Becker et al [14] developed a novel clustering approach called "Overlapping Clustering Generator" (OCG) which can be described as follows: (i) finding all

centered cliques, clusters, using a greedy polynomial approach; (ii) computing the modularity of all clusters; (iii) combining the clusters  $c_i$  and  $c_j$  whose maximal gab, defined by the difference between the modularity values of each cluster, is positive and iterating this operation until either the expected number of clusters or the maximum number of nodes in a cluster is reached; and (iv) enhancing the modularity values as well as the performance of the developed algorithm, OCG, by transferring each protein to the clusters where its contribution maximizes the modularity value of the clusters.

Liu *et al* [25] developed an approach for clustering PPI networks called **ADHOC** which based on a new subgraph density metric. First, for each vertex, ADHOC computes the degree, clustering coefficient and the local-density coefficient, MinCC, values for each node. There is a well-known clustering coefficient formula but they developed a new local-density measurement method, MinCC, by including the degree of the vertex as a significant parameter into the clustering coefficient. Based on the following parameters  $k, d$  and  $MinCC$  values, ADHOC method can be summarized in the following steps: (i) grouping the set of nodes in the studied graph into four types: (1) density nodes which have clustering coefficient values greater than or equal to their MinCC values and the density region is defined as the set of adjacent nodes of the density nodes except the nodes which are not connected to other neighbors (2) every node in the density region and not density node is called border node. (3) affiliated node to a cluster is a node whose edges are connected to the clusters nodes. (4) interspersed nodes are all the

remaining nodes in the studied graph; (ii) after classifying all the nodes, grouping the density nodes that are directly connected as well as the border nodes that are directly connected to those density nodes to the same cluster while the affiliated nodes are assigned to the clusters that are connected with.

Bader and Hogue [11] proposed a density-based algorithm called "Molecular Complex Detection" (MCODE). First, MCODE assigns weights to all nodes which are computed based on the core clustering coefficient. Second, starting with a cluster  $c$  of size one which contains the node with the highest weight and iterates to include all nodes that are neighbors to the nodes in the cluster  $c$  and have weight above a given threshold  $\tau$ , and this continues until all nodes have been checked. Finally, it removes every obtained cluster  $c$  that contains one node.

ClusterOne approach proposed by Nepusz et al [12] is another recent algorithm for finding overlapping clusters in PPI network. It is similar to MCODE. ClusterOne is an agglomerative method starting from a single seed vertex, and adds or removes vertices greedily to find groups with high cohesiveness. Then, it merges each pair of groups where the overlap score is above a specified threshold. Finally, it removes all clusters of size less than three vertices or whose density is below a given threshold.

Rhrissorrakrai and Gunsalus [26] extends MCODE [11] approach by proposing an algorithm called MINE to identify overlapping clusters in biological networks. In MINE, the weight of each vertex  $v$  is initialized with a value obtained by the multiplication of the clustering coefficient value of  $v$  with respect to the cluster

involved the vertex  $v$  and the highest degree in the  $v$ 's neighbors  $N[v]$ . Moreover, the modularity of a cluster is defined as the ratio between the size of the intra-cluster and the size of the inter-cluster.

Another overlapping clustering method is clique percolation method (CPM) developed by Palla *et al* [28]. CPM consists of two main phases: (i) based on greedy concept, all maximal cliques of a given size  $k$  in the considered network are identified, where  $k$  takes values between  $s$  (the largest degree over all vertices) and 2, (ii) constructing a clique-clique overlap symmetric matrix in which rows and columns represent cliques and its entries are the shared nodes between the corresponding two cliques. A cluster is defined to be all  $k$ -cliques that share  $k - 1$  nodes. An application called CFinder which implemented by Adamsek *et al* [13] uses the CPM approach.

Zhang *et al.* [29] applied CPM [28] on a line graph  $L(G)$  which is obtained from the original graph  $G$  represented a PPI network. Then, the clusters discovered in  $L(G)$  are transformed back to groups in  $G$ . Finally, any pairs of clusters that are heavily overlapped are merged to one cluster.

Cho *et al* [27] proposed an information flow approach for finding overlapping functional groups in PPI networks. The proposed algorithm can be described as follows: (i) assigning a weight to each node as follows: the weight of a vertex  $v$  is the summation of its incident edges' weights which are computed by using Pearson's correlation measure; (ii) picking up set of nodes (informative nodes) having the highest weights which correspond to the preliminary number of

obtained clusters in the studied network; *(iii)* identifying preliminary clusters by considering each node  $s \in \{ \textit{informative nodes} \}$  as starting point which expanded to include all its reliable neighbors (the edge between a pair of nodes has a positive weight) and iterating this process until all nodes in the network are clustered.

### 3.3 Model-Based Methods

Actually, real complexes in the organism is not limited to densely connected subgraph but parsley connected subgraphs are also existed in PPI networks. Since density-based algorithms usually neglect the proteins that connect with main complexes by few edges even though these proteins may represent primary interaction, it is crucial to develop methods to identify overlapping complexes and complexes that covers peripheral proteins with low density.

Zhang et al [16] argue that the density-based approaches cannot identify the sparse complexes as well as the proteins that have a few connections to dense complexes. They developed a method called "regularized sparse generative network model" (RSGNM) for finding protein communities in PPI network. This method can discover the sparse and dense subnetworks. They rely on the observation that two proteins that have higher propensities, which specifies the likelihood that proteins belong to some modules, may interact with each other. The developed algorithm can be outlined as follows: *(i)* finding exclusive clusters using a SPICi algorithm [10]; *(ii)* based on  $K$ , the number of clusters obtained from SPICi method, a protein-complex indication matrix  $\hat{F}$  is constructed where its rows rep-

resent the nodes (proteins), columns represent the clusters and its elements either 0 or 1 e.g.  $\hat{f}_{i,z} = 1$  if protein with index  $i$  belongs to the module with index  $z$  otherwise 0; (iii) initializing the propensity matrix; (iv) using a multiplicative update rule to optimize the obtained clusters and results the protein-complex indication matrix  $F^*$  which shows all proteins in the studied network and the complexes to which belong using  $f_{i,z} = 1$  if  $f_{i,z} \geq \tau$  (they give 0.3 to the threshold). The propensity matrix  $F^*$  indicates the number of complexes and each protein to which complexes belongs. Such method based on a lot of parameters and it also based on a multiplicative update rule which needs a lot of time especially where the data is very large.

Zhang *et al* [17] proposed another approach called Regularized Sparse Random Graph Model, *RSRGM*, for detecting cohesive, non-cohesive and overlapping complexes in PPI network. This method actually extended to the previous approach RSGNM [16] with the following modifications: instead of using a method to find the exclusive clusters, they initialized the protein-complex indication matrix  $\theta$  and the maximum number of possible functional groups,  $K$ , randomly. Then, the same RSGNM [16] steps are used in RSRGM.

### 3.4 Random Walk-based Methods

Dongen [9] proposed a Markov clustering method (MCL) which based on random walks (called flow) within a graph. MCL partitions the graph into clusters by applying two alternative operators: (1) expansion operator which defined by cal-



culating successive powers of the associated transition matrix  $M$  using the normal matrix product (i.e. matrix squaring) in order to allow flow to connect different regions of the graph, (2) Inflation operator is defined by raising each single column to a non-negative power, and then re-normalizing in order to further strengthen the cohesive regions and demote the sparse regions. Although MCL is very efficient and scalable, it has the drawback that it partitions the graph into multiple cohesive exclusive clusters.

### 3.5 Genetic Algorithm-Based Methods

Anirban et al [19] proposed an algorithm (PROCOMOSS) to detect overlapping clusters in PPI network using genetic algorithm technique. They rely on the properties captured in the graph modeling the PPI network and they also utilize the GO terms to consider the biological properties of the proteins. Their approach can be described as follows: First, encoding the chromosome as a vector of integer numbers representing the indices of the proteins in the proteins set. Then, initializing the population based on applying k-means clustering on both dimensions of the adjacency matrix  $A$  of a graph modeling PPI network. Next, calculating the fitness values of each individual of the population using two objective functions. Finally, selecting parents by adopting the same way used in NSGA-II [30] and mutating the selected chromosome as follows: select a random node and then either remove that node or add its neighbors to the selected chromosome with the same probability. The main drawback of this approach is that the predicted

clusters cover a small percentage of the PPI network as well as this algorithm uses NSGA-II [30] which its complexity is  $O(MN^2)$ , where  $N$  be the size of the population and  $M$  is the number of objectives.

# CHAPTER 4

## A NEW GA BASED CLUSTERING APPROACH

### 4.1 Research Problem

As shown in Chapter 3, many density-based clustering approaches compute the density for each vertex on the basis of different density, modularity and clustering coefficient measures. Then, they always start from a seed (vertex with the highest weight) and expand to include the other vertices in order to cluster the network according to greedy strategy. Such approaches discard a lot of nodes having low weights; the predicted clusters are not highly directed to each other, i.e., the overlapping degree distribution is low since the PPI is a disassortative network in which the highly weighted vertices are not directly linked to each other while, in nature, the nodes (proteins) can be involved in several protein complexes. Furthermore, being an optimization technique, starting from a set of candidate

solutions is much better than a single solution which based on a greedy procedure as the most optimal short-term solution may lead to the worst potentially long-term results.

Chapter 3 also shows that there is only one GA-based clustering approach in the PPI literature. Such method discards a lot of proteins from the original PPI network since the inappropriate representation of the chromosome used. It uses semantic similarity measure to compute the fitness value for each possible solution in the population and the parents are selected using binary crowded tournament selection method which are very demanding complexity.

As essential features of GA-based clustering method in the context of PPI are that the chromosome representation should take into account the overlapping property and the variety of the clusters size in order to be close to the original clustering solution; the variety among solutions in the first population should be very high in order to prevent the premature convergence; and the most important characteristic of GA based clustering algorithm that the fitness function must be well-defined and capable of finding optimal solution in which the clusters should contain more internal links among nodes inside the cluster than external links to other clusters.

Guided by the previous issues (chromosome representation, population initialization and fitness function definition) we have developed a clustering approach on the basis of GA technique that takes into account the main characteristics of PPI networks (multifunctionality, scale-free structure, small-world property and

disassortativity) to perform better than existing clustering algorithms.

## 4.2 Research Approach

### 4.2.1 Introduction

In this section we present an overlapping clustering approach to identify protein complexes in protein-protein interaction networks.

Algorithm 2 provides the high-level description followed in our study for clustering the PPI network. Starting with initial population of individuals (set of clusterings), the algorithm generates individuals using genetics operators (i.e., selection and mutation). The goal is to get individuals to converge to solutions (clusterings) of maximum fitness according to the objective function.

---

**Algorithm 2** Clustering Algorithm high-level description.

---

- 1: Population initialization.
  - 2: **while** Number of generations limit has not been exceeded **do**
  - 3:     Evaluate fitness of all individuals of the current generation population.
  - 4:     Select survivals to next generation.
  - 5:     Mutate survivals.
  - 6: **end while**
-

### 4.2.2 Representation and Initialization

Before using GA, we have to represent its chromosome appropriately, defining the blueprint of a possible solution. Since the result of clustering problems is a set of overlapping clusters each with different size (the size of the cluster is the number of proteins belongs to it), such representation should be as close as possible to the original one. Anirban et al [19] encoded the chromosome as a list of  $n$  integer number. Thus, the population includes  $m$  lists (clusters). Consequently, in such representation, the number of predicted clusters is highly correlated to the population size; the size of the clusters is very high; and the overlapping degree distribution is also high. All those issues resulted in the high discarded percentage of proteins in the original network.

In the clustering social networks literature, Blas et al[31] represented the chromosome as a list including two parts. The first part is of length  $N$ , where  $N$  is the size of the network, while the second part involves  $m$  integer numbers in the range  $\{1, \dots, k\}$ , where  $k$  represents the number of clusters. In such representation, the value of the element  $j$  in the first part of the list represents the cluster to which  $j$ th node is assigned. Consequently, each node is assigned to a single cluster which is inappropriate representation to clustering problems in the context of PPI.

Tasgen and Bingol[32] represented each chromosome as an array of  $n$  integer numbers, where  $n$  is the number of nodes on the considered network, and each element  $j$  in the array represents the cluster to which  $j$ th node is assigned which is also inappropriate representation to partitioning PPI networks.

In this study, we represent each chromosome (individual) as  $k$  lists  $\{c_1, c_2, c_3, \dots, c_k\}$ , where  $k$  is the number of clusters. Each list can store integer numbers in the range  $\{1, 2, \dots, N\}$ , where  $N$  is the size of the data set. The element  $j$  of a list is a node's index of the graph  $G$  modeling the PPI network. It is possible that some elements of different lists can hold the same value  $j$  which means that a protein with index  $j$  can exist in more than one cluster; this is in case of overlapping clustering.

$c1:$	1	20	8	70	400	...
$c2:$	12	220	...	8	200	...
$c3:$	400	5	...	1	30	90
$ck:$	200	120	1000	...	400	

Figure 4.1: Chromosome representation [for our clustering approach].

Once the blueprint of the possible solution is determined, we create the first population which composed of a number (population size) of individuals, possible clusterings. We use two different methods to initialize the population. The first approach, generating  $m$  random individuals, where  $m$  is the size of the population, as follows: for each individual consisting of  $k$  lists, assigning an integer value  $j$  in the range  $\{1, 2, \dots, N\}$  where  $N$  is the size of data set for each element randomly. For example, as illustrated in Figure 4.2, the node with index 70 is assigned to the cluster  $c_1$  while the node with index 8 is assigned to two clusters  $c_1$  and  $c_2$ . Such a way should take into account the variety among the individuals of the population which supposed to be considerably high to prevent the tendency to a premature convergence (failing in local optimal solution which is an optimal within a neighboring set of feasible solutions).

---

**Algorithm 3** Generating an individual randomly.

---

- |   |  |
|---|--|
| 1: Define a vector $L$ containing a random permutation of the integers from 1 to $N$ inclusive. | ▷ Let $N$ be the size of the network.        |
| 2: Divide $L$ to $k$ parts.   | ▷ Let $k$ be the number of Clusters.         |
| 3: Assign each part of $L$ to a cluster $c_i$ .   | ▷ Let $i$ be in the rang $\{1, \dots, k\}$ . |
-



$c1:$	1	20	8	70	400	...
$c2:$	12	220	...	8	200	...
$c3:$	400	5	...	1	30	90
$ck:$	200	120	1000	...	400	

Figure 4.2: Population initialization method [using the random initialization method and spectral clustering method].

The second way, we use the resulting complexes of spectral clustering algorithm [33] to create the initial population. This method can be described as follows: given an adjacency matrix  $A$ . First, construct the degree matrix  $D$ , the degree for each vertex is computed by the number of adjacent vertices of that vertex  $d_i = \sum_{j=1}^n A_{ij}$ . Then, compute the symmetric Laplacian matrix  $L$ . Next, identify  $v_1, v_2, \dots, v_k$  the top  $k$  eigenvectors of  $L$ . Then, construct the matrix  $V \in \mathbf{R}^{n \times k}$  from  $v_1, v_2, \dots, v_k$ . Finally, each row of  $V$  represents a vertex in  $\mathbf{R}^k$ , group these vertices into  $k$  clusters using any approach such as *k-means* algorithm. In such a case, a set of exclusive clusters is predicted which used to initialize the population.

Once the population is initialized, the algorithm performs the genetic algorithm operations for a number of iterations called *generation*. These operations are discussed in details in the following subsections.

### 4.2.3 Objective Function

The objective function aims to calculate the fitness values for each individual of the population to indicate how well each individual is suited to be the solution of a given problem. In Table 4.1, we present the objective functions used to compute the fitness values for population in the literature.

Table 4.1: The objective functions used in previous published works.

Authors (Year)	Objective Function	Network Type	Details
Pizzuti [34] (2008)	$F = \sum_i^k Q(S_i)$	Social works	Net- $Q(S_i) = M(S_i) \times vs$ is the fitness value for each cluster $S_i$ , where $vs$ is the number of 1's in the adjacency matrix $A(I, J)$ representing the cluster $S_i$ and $M(S_i)$ is the average of $S_i$ to the power $r$ , $M(S_i) = \frac{\sum_{i \in I} (a_{ij})^r}{ I }$ .
Tasgen and Bingol[32] (2007)	$F = \sum_i (e_{ii} - a_i^2)$	Different Com- plex Networks	where $e_{ii}$ is the cluster size; and $a_i$ is the number of edges that has an endpoint in the cluster $i$ to the total number of links in the network.
Anirban et al [19] (2012)	$f_1 = \frac{E}{N(N-1)}$  $f_2 = \frac{\sum_{i \in p} \sum_{j \in p} s(i,j)}{p}$	PPI Network	This method based on two fitness functions $f_1$ and $f_2$ , where $f_1$ is the fitness value of a cluster $C$ ; $E$ is the number of edges in the cluster $C$ ; and $N$ is the number of nodes in $C$ .  where $S$ is the similarity matrix of each pair of proteins; $s_{i,j}$ is an element of the matrix $S$ ; and the similarity matrix is constructed using three semantic similarity measures proposed by Lin [35], Jiang and Conrath [36], and Kappa [37].

In our case, the fitness value of an individual reflects the intra-cohesion of each cluster proposed by the individual as well as the inter-cluster coupling of those clusters. The goal is to maximize intra-cohesion and minimize inter-coupling. We represent intra-cohesion and inter-coupling by the number of edges within and across clusters, respectively. In this thesis, we designed a new fitness function and we also used three objective functions [33] proposed in the literature to capture the goodness of a partition of the networks and to evaluate the quality of possible cluster structures. Here, GA is to optimize those objective functions in order to find an optimal solution. The clusters in the potentially good solution should contain more internal links among the nodes inside the clusters than external links to other clusters. We compare the clustering achieved using these objective functions to the one achieved by our proposed objective function. We also compare clustering of all four objective functions to MIPS and CYC2008.

- Ratio cut objective function:

$$R_{cut}(C_1, \dots, C_k) = \sum_{i=1, j \neq i}^k \frac{|C_i|}{W(C_i, C_j)} \quad (4.1)$$

where  $k$  is the number of clusters,  $|C_i|$  is the number of nodes in the cluster  $C_i$  and  $W(C_i, C_j)$  is the number of edges that has just one endpoint in the cluster  $C_i$ .

Ratio cut based on the size of the clusters which is the number of vertices in the cluster. Assume the clusters of an individual look like the cluster shown in figure 4.3. According to ratio cut measure such individual gets high value,

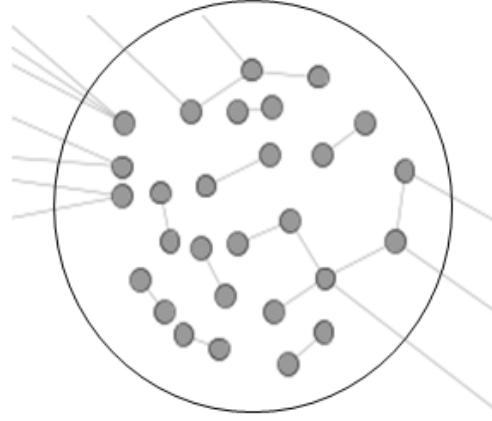


Figure 4.3: A Cluster representing ratio cut limitation.

though its clusters contain many separated nodes and many disconnected components.

- Normalized cut objective function:

$$N_{cut}(C_1, \dots, C_k) = \sum_{i=1, j \neq i}^k \frac{Vol(C_i)}{W(C_i, C_j)}. \quad (4.2)$$

where  $Vol(C_i)$  is the degree of every node in the cluster  $C_i$ . Normalized cut

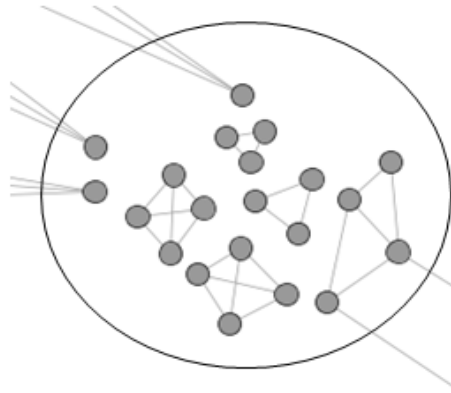


Figure 4.4: A Cluster representing normalized cut limitation.

based on the volume of the clusters which is the the degree of each vertex

in the cluster. Assume the clusters of an individual look like the cluster shown in figure 4.4. According to normalized cut measure such individual gets high value, though its clusters contain many separated nodes and many disconnected components.

- Min-Max-cut objective function

$$M_{cut}(C_1, \dots, C_k) = \sum_{i=1, j \neq i}^k \frac{W(C_i, C_i)}{W(C_i, C_j)}. \quad (4.3)$$

where  $W(C_i, C_i)$  is the number of edges inside the cluster  $C_i$ . As shown

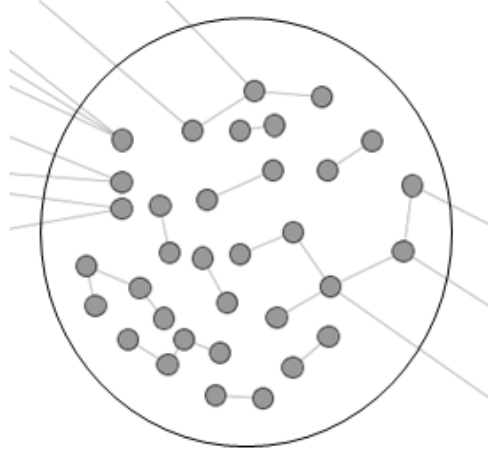


Figure 4.5: A Cluster representing max-min cut limitation.

in figure 4.5, the same issue in the Min-Max-cut objective function, if the number of internal links are much more than the external links the cluster get high fitness value even though it contains many disconnected components.

We designed a new fitness function to compute the fitness value for each individual

as follows:

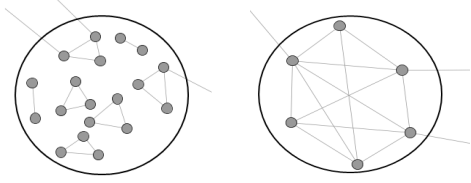
$$D_{cut}(C_1, \dots, C_k) = \sum_{i=1, j \neq i}^k \frac{W(C_i, C_i)}{A_i + W(C_i, C_j)} \quad (4.4)$$

where  $W(C_i, C_i)$  is the number of edges inside the cluster  $C_i$ ,  $W(C_i, C_j)$  is the number of edges that has one endpoint in  $C_i$  and  $A_i$  is the maximum possible number of edges in the cluster  $C_i$ .

We added the term  $A_i$  to make sure we are pushing for maximizing the cluster cohesion and  $W(C_i, C_j)$  to make sure we are pushing for minimizing the cluster coupling. Thus, according to this measure, any groups with high value represent a good clustering because they are well-connected to each other and sparse connected to the rest of the network. The comparative results of those four objective functions performance is discussed in Chapter 5.

Table 4.1 shows an example of the four objective functions used in this study. It is clearly that the proposed fitness function (density cut) capture the goodness of a cluster better than the others. Based on density cut, cluster (b) gets higher value than cluster (a) as it is more cohesive and does not include neither separated nodes nor disconnected components. On the other hand, ratio cut, normalized cut and max-min cut give higher values for the cluster (a) even though it is sparse and involves many disconnected components

Table 4.2: An illustrative example of the four objective functions used.



Objective Function	cluster (a)	cluster (b)
Density Cut $\frac{E_{in}}{A_i +  E_{out} }$	$\frac{17}{17+3} = 0.10$	$\frac{11}{11+3} = 0.61$
Ratio Cut $\frac{ C_i }{ E_{out} }$	$\frac{19}{3} = 6.3$	$\frac{6}{3} = 2$
Normalized Cut $\frac{Vol(C)}{ E_{out} }$	$\frac{37}{3} = 12.3$	$\frac{25}{3} = 8.3$
Max-Min Cut $\frac{E_{in}}{ E_{out} }$	$\frac{17}{3} = 5.7$	$\frac{11}{3} = 3.7$

#### 4.2.4 Genetic Operators

The most common operations used in genetic algorithm are selection, crossover and mutation. Here, we exclude the crossover operation as it resulted in too much exploration and disturbed the exploiting potentially good solutions. Regarding the parent selection defined as the process of selecting individuals from the current population to create offsprings for the next generation. This process aims to emphasize that the individuals with high fitness values are chosen in hopes that their offsprings will have higher fitness as well. There are many ways to select parents, individuals, from the current population for reproduction. Algorithm 4 illustrates in detail the parent selection method used.

---

**Algorithm 4** Selection Process.

---

```
1: sort the individuals according to their fitness values.
2: select  $n$  individuals called - elite parents - having the highest fitness values to
   the next generation without mutation, we set the elite rate to 0.20.
3: calculate the cumulative sum  $S$ , of all the individuals' fitness values.
4: for  $N$  times do     $\triangleright$  let  $N$  be the size of the population minus the number of
   the elitism parents.
5:     generate a real random number  $r$  between 0 and  $S$ .
6:
7:     while  $s < r$  do
8:         go through the population and summing cumulative values.
9:     end while
10:    select the individual corresponding to the cumulative sum value  $s$ .
11: end for
```

---

Mutation operation is defined as performing some changes in the values of a specific chromosome, individual. Consequently, the GA may reach to a better solution with the obtained individuals. We adapt the mutation operator used in [19] and modify it in such a case to be suited and more efficient to our problem. This operation can be described as follows: after selecting an individual to be mutated, its nodes are either moved from one cluster to another as shown in Figure 4.6 or some nodes of the graph  $G$  are added to the selected individual as shown in Figure 4.7. Algorithm 5 illustrates in detail the mutation operator used.



---

**Algorithm 5** Mutation Process.

---

```
1: for  $n$  times do       $\triangleright$  Let  $n$  be the number of clusters in the selected parent.
2:   generate a real random number  $r_1$ .
3:   if  $r_1$  is less than the mutation rate (0.4) then
4:     for  $N$  times do       $\triangleright N$  is the number of changes.
5:       generate a real random number  $r_2$  between 0 and 1.
6:
7:       if  $r_2$  is less than a threshold  $\tau$  then
8:         move a random selected node from the cluster  $c_i$ 
9:         to another cluster  $c_j$  as illustrated in Figure 4.6
10:      else
11:        add the adjacent nodes of the selected node
12:        to  $c_i$  as shown in Figure 4.7.
13:      end if
14:    end for
15:  end if
16: end for
```

---

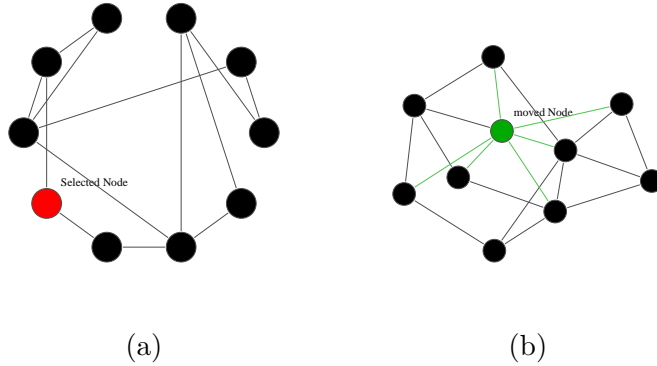


Figure 4.6: Mutation operation. (a) shows the selected node of the cluster  $c_i$ . (b) shows the cluster  $c_j$  after the mutation operator.

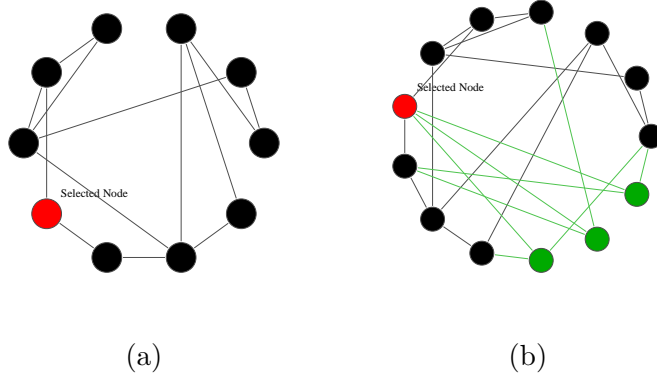


Figure 4.7: Mutation operation. (a) shows the selected node of the cluster  $c_i$ . Figure (b) illustrates the cluster  $c_i$  after adding the selected node's neighbors from the graph  $G$ .

# CHAPTER 5

## EXPERIMENTS AND RESULTS

### 5.1 GA parameters setup and optimization

In using GA, the parameters of GA must be initialized in advance. Table 5.1 shows the values for all GA parameters used in our clustering approach. Those values were selected subjectively as follows: initially, we follow the mostly used values according to the previous published works [19][34][38] to initialize the population size, number of generations, mutation rate and elitism rate. Then, we gradually refined such values in the subsequent experiments according to the feedback reported from the preceding experiment.

Table 5.1: GA parameters setup using four different objective functions to compute the fitness values of the population.

No	Parameter	$D_{cut}$	$M_{cut}$	$N_{cut}$	$R_{cut}$
1	Population Size	50	50	50	50
2	Chromosome Size	200	200	200	300
3	No. of Generations	30	30	30	30
4	Mutation Rate	0.4	0.4	0.4	0.4
5	Elitism Rate	0.2	0.2	0.2	0.2

## 5.2 Results Analysis

### 5.2.1 Data Set

We study protein interaction network from yeast organism since there are abundant high-confidence data sets for the yeast PPI network as well as there are high-confidence reference complex sets. In our experiment, we applied our clustering algorithm on the Collins PPI network [39] extracted from BioGrid data set. This network has 8319 interactions among 1004 proteins. It has an average degree (16.57) where the degree of a node in a network is the number of links connected to the node; the density of this network is 0.016 (density is ratio between the total number of connections and the potential connections that can exist in the network). In order to validate the resulted clusters whether they have any biological meaning we use two common approaches: (i) using two hand-curated gold-standard complex sets: CYC2008 [40] which includes 408 protein complexes and MIPS [41] catalog consisting of 203 protein complexes and (ii) using cellular components from GO terms. We present the clusters validation in the following subsections.

### 5.2.2 Cluster validation based on known complexes

We use three quantity measures: precision, recall and F-score to evaluate the performance of different clustering algorithms in terms of the similarity rate between the identified clusters and a set of validation protein complexes derived from CYC2008 and MIPS catalogs. For each predicted cluster  $C$ , let true positive (TP) be the set of proteins shared between the cluster  $C$  and a reference protein complex  $G$  while false positive (FP) is defined as the set of proteins existed only in the cluster  $C$  and true negative (TN) is defined as the proteins that are members of the reference complex  $G$  but not found in the cluster  $C$ . Hence, Recall, precision and F-measure scores are calculated according to the following equations:

$$Recall = \frac{TP}{TP \cup TN} \quad (5.1)$$

$$Precision = \frac{TP}{TP \cup FP} \quad (5.2)$$

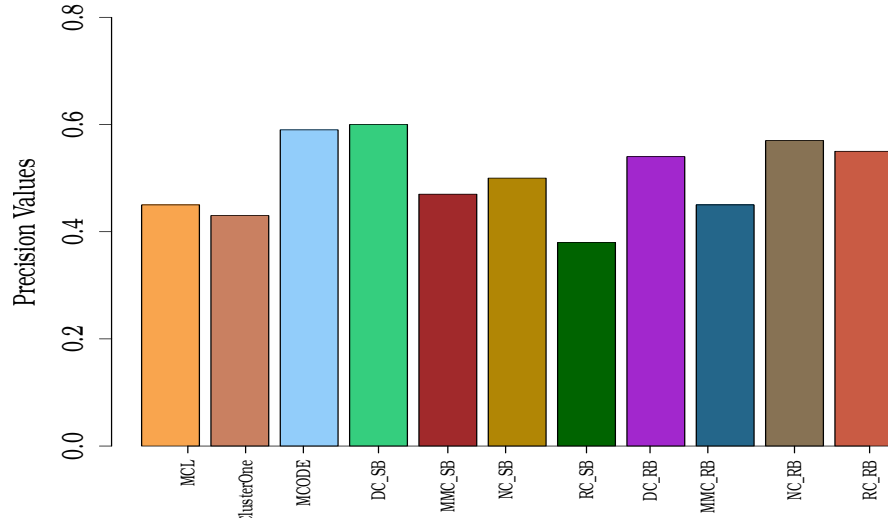
$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (5.3)$$

As stated in Chapter 4, we use two different ways to create the initial population while four objective functions are used to calculate the fitness values for each individual. In order to assess the performance of the proposed clustering method, we compared our clustering approach to three competing clustering algorithms: one exclusive clustering method (MCL [9]) and two overlapping clustering approaches (MCODE [11] and ClusterOne [12]). First, we employed the previ-

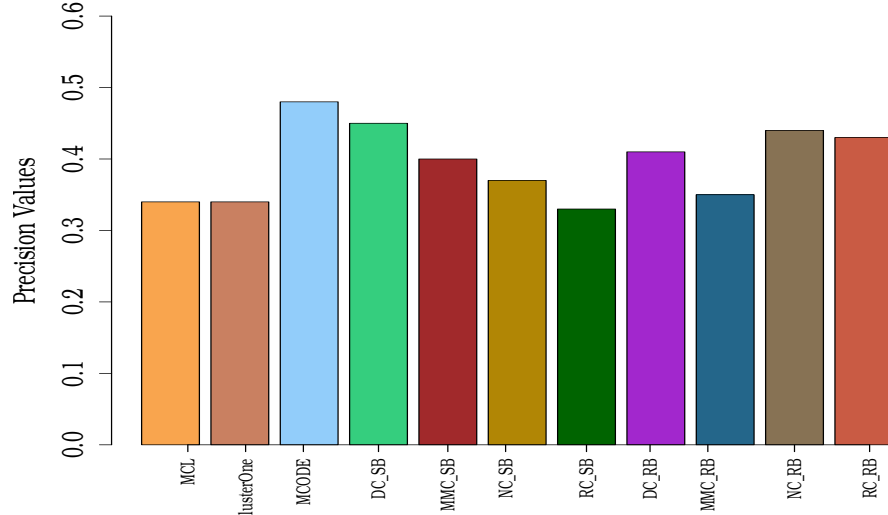
Table 5.2: Comparison of Clustering Algorithms on the Yeast Collins Network.

Method	Obj. Functions	#Cls	CYC2008				mips		Discard
			Recall	Precision	F-measure	Recall	Precision	F-measure	
MCODE		54	0.66	0.59	0.63	0.27	0.48	0.35	40%
MCL		75	0.65	0.45	0.54	0.27	0.34	0.30	19%
ClusterOne		114	0.55	0.43	0.49	0.20	0.34	0.25	18%
Spectral	1) Density cut	162	0.74	0.60	0.66	0.32	0.45	0.37	14%
	2) Maxmin cut	180	0.71	0.47	0.60	0.38	0.40	0.39	15%
	3) Normalized cut	193	0.67	0.50	0.57	0.39	0.37	0.37	20%
	4) Ratio cut	161	0.73	0.38	0.50	0.39	0.33	0.36	17%
Random	5) Density cut	164	0.72	0.54	0.62	0.30	0.41	0.35	18%
	6) Maxmin cut	162	0.71	0.45	0.56	0.40	0.35	0.38	17%
	7) Normalized cut	138	0.66	0.57	0.61	0.36	0.44	0.41	19%
	8) Ratio cut	154	0.61	0.55	0.58	0.34	0.43	0.38	18%

ous validation measures on each predicted cluster resulting from the considered algorithms. Then, the averaged value for each measure are calculated and reported. Table 5.2 shows the overall results of the comparison according to the three evaluation scores: recall, precision and F-measure and using two reference protein complexes CYC2008 and MIPS. In general, as also graphically shown in Figures 5.1-5.3, among the considered algorithms, we note that none of these methods surpasses all the others in terms of the three validation scores and using CYC2008 and MIPS reference complexes. To summarize, our method which based on density cut objective function outperforms MCL and ClusterOne methods on the three validation scores on both CYC2008 and MIPS reference sets. On the other hand, although our method which based on the clusters resulting from spectral algorithm to initialize population and using ratio cut as an objective function outperforms all the others in terms of recall score using CYC2008 reference set, it obtains lower precision and f-measure values compared with the other approaches using CYC2008 and MIPS complexes. MCODE also outperforms all the other methods in terms of precision metric, but it predicts a fewer number of clusters and discard a high proportion of the proteins in the original network compared with the other approaches.



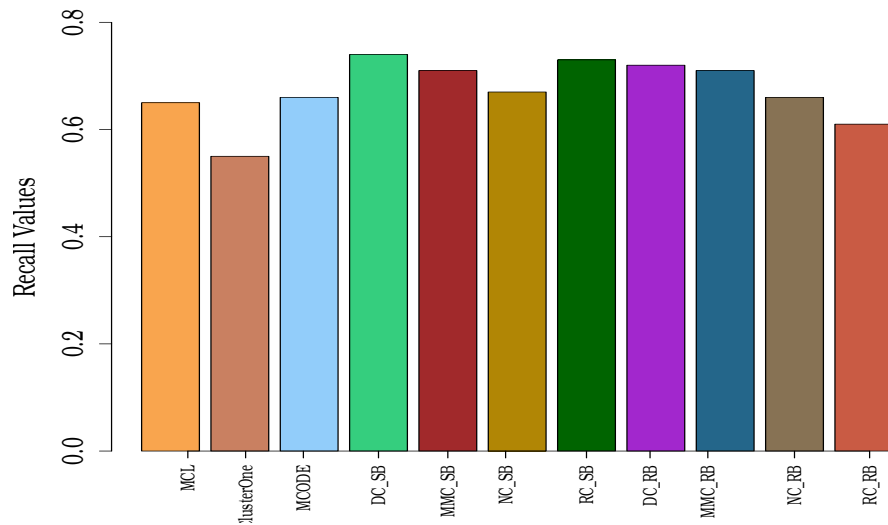
(a) precision values(CYC2008 reference set.)



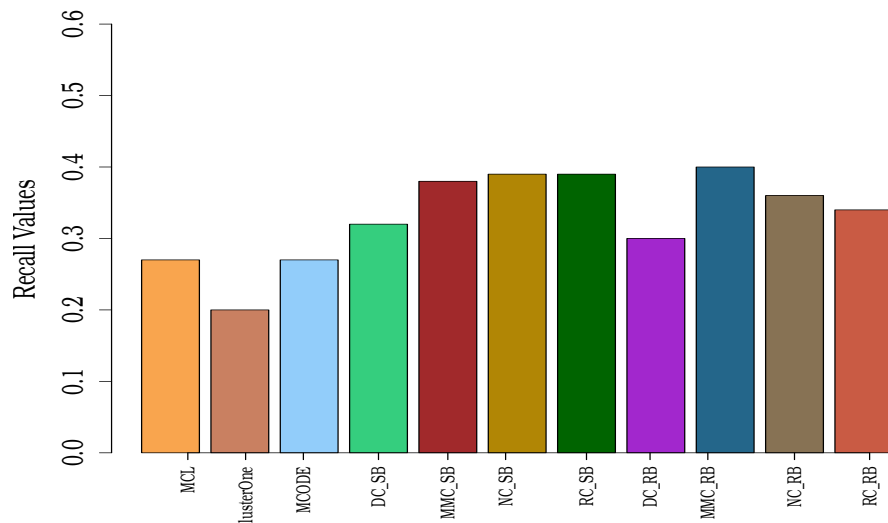
(b) precision values(MIPS reference set.)

Figure 5.1: Comparative results of the considered clustering approaches using Precision measure.



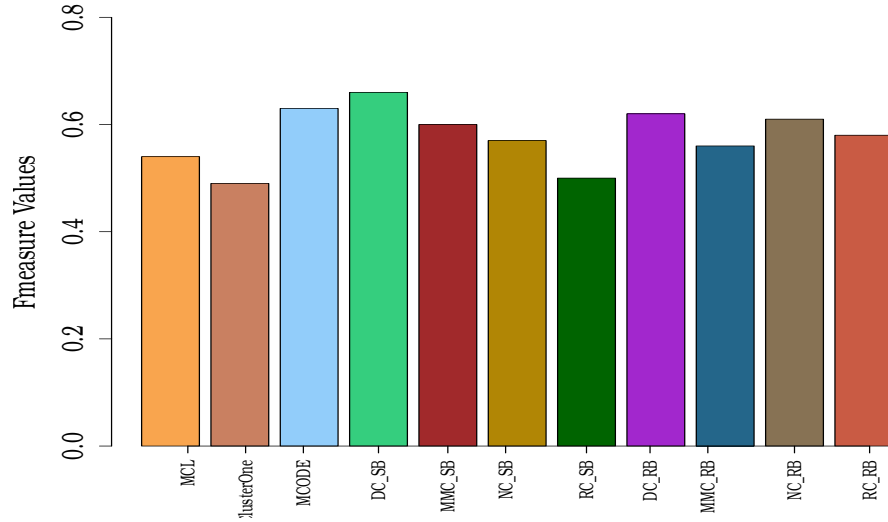


(a) recall values(CYC2008 reference set.)

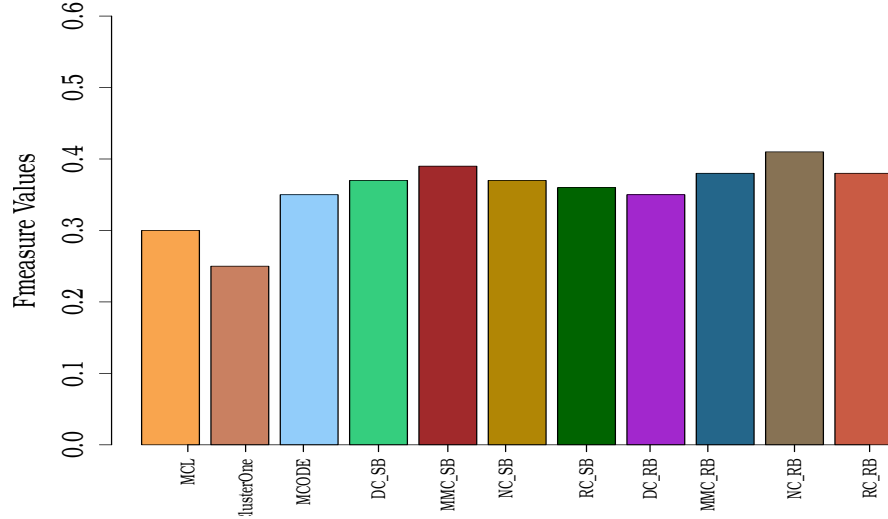


(b) recall values(MIPS reference set.)

Figure 5.2: Comparative results of the considered clustering approaches using Recall measure.



(a) f-measure values(CYC2008 reference set.)



(b) f-measure values(MIPS reference set.)

Figure 5.3: Comparative results of the considered clustering approaches using f-measure.

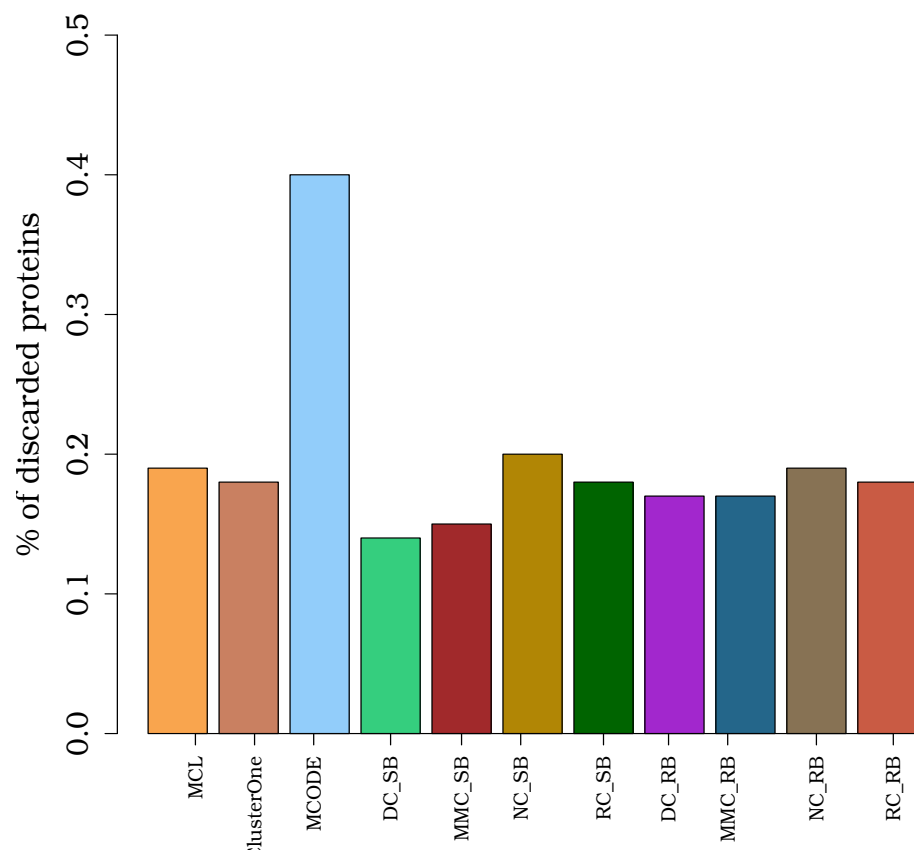


Figure 5.4: The percentage of discarded proteins in the Collins network.

We also illustrate the number of complexes and the percentage of the proteins covered in the predicted clusters resulted from all the considered algorithms in Table 5.2. It is obvious that our clustering method usually discovers more complexes while MCODE predicts fewer complexes since it tends to search for a high density clusters. And the other approaches, MCL & ClusterOne, predict fewer modules than our methods and more complexes than MCODE. Regarding the percentage of the proteins covered in the predicted clusters, it is obvious that our method which based on the clusters resulting from spectral algorithm to create initial population and using density cut objective function outperforms all the other approaches and gets the lowest percentage of the discarded proteins as shown in Figure 5.4; high value of coverage indicates that a high proportion of the proteins in the considered PPI network are clustered. On the other hand, MCODE algorithm obtains the highest percentage of discarded proteins. So, a high percentage of proteins are lost in the clustered network. Regarding the other methods, the percentage of covered proteins is almost similar, meaning that the same proportion of proteins in the original network are assigned to the clusters.

We evaluated the distribution of the recall, precision, f-measure and the percentage of discarded proteins values over many experiments. Actually, once we selected the values of GA parameters, as shown in Table 5.1, we run our algorithm using four fitness functions 40 times (10 times for each fitness function) and we calculated recall, precision, f-measure and the percentage of discarded proteins in each experiment. Then, we evaluated the distribution of the results. Table 5.4 and Table 5.3 show the values of some statistical measures used, as shown in those tables, the proposed fitness function (density cut) performs better than the others and it got good recall, precision and f-measure average values. Moreover, it got low percentage average value of the discarded proteins. The standard deviation values are very small which are good and mean that the values of those measures in each experiments are close to the average values.

Table 5.3: The average, standard deviation, max and min of the recall, precision and f-measure validation scores used to validate the resulted clusters of 5 runs[the 1st population of GA is generated randomly].

Fitness Function	Measure	Recall	Precision	F-measure	Discard
Density Cut	Mean	0.72	0.58	0.65	%12
	Standard deviation	0.01	0.03	0.02	0.03
	Max	0.74	0.61	0.67	0.18
	Min	0.71	0.54	0.61	0.10
Max-Min Cut	Mean	0.70	0.57	0.63	%18
	Standard deviation	0.02	0.07	0.04	0.05
	Max	0.73	0.62	0.66	0.26
	Min	0.67	0.45	0.55	0.11
Normalized Cut	Mean	0.69	0.58	0.63	%17
	Standard deviation	0.02	0.02	0.02	0.03
	Max	0.71	0.60	0.65	0.21
	Min	0.66	0.56	0.61	0.14
Ratio Cut	Mean	0.67	0.57	0.62	%14
	Standard deviation	0.04	0.03	0.03	0.03
	Max	0.72	0.61	0.66	0.18
	Min	0.61	0.52	0.58	0.10

Table 5.4: The average, standard deviation, max and min of the recall, precision and f-measure validation scores used to validate the resulted clusters of 5 runs [the 1st population of GA is generated using the clusters resulting from spectral clustering algorithm].

Fitness Function	Measure	Recall	Precision	F-measure	Discard
Density Cut	Mean	0.74	0.56	0.64	%07
	Standard deviation	0.01	0.03	0.02	0.04
	Max	0.76	0.59	0.66	0.14
	Min	0.73	0.53	0.61	0.05
Max-Min Cut	Mean	0.69	0.51	0.58	%14
	Standard deviation	0.02	0.03	0.02	0.05
	Max	0.71	0.55	0.61	0.21
	Min	0.67	0.48	0.57	0.08
Normalized Cut	Mean	0.68	0.53	0.59	%14
	Standard deviation	0.01	0.03	0.02	0.04
	Max	0.69	0.56	0.62	0.19
	Min	0.67	0.50	0.57	0.10
Ratio Cut	Mean	0.67	0.49	0.56	%11
	Standard deviation	0.06	0.06	0.04	0.04
	Max	0.74	0.55	0.61	0.16
	Min	0.61	0.38	0.50	0.06

Figure 5.5 describes the best fitness values within 50 generations for arbitrarily chosen run. We observe that the population converges more quickly in the case with density cut objective function (using two methods, random and spectral clustering, to create the first population) and ratio cut objective function (using spectral clustering method to initialize the population) than ratio objective function (using random method to initialize the population), normalized objective function and max-min objective function. To be clarified, we can not increase the number of generations because, during the experiments, we noticed that when using more than 50 generations the approach is occasionally resulted in too much exploration of the search space which leads to produce clusters with large size and high overlapping degree distribution (the overlapping degree of a cluster is the number of other clusters that share common proteins).

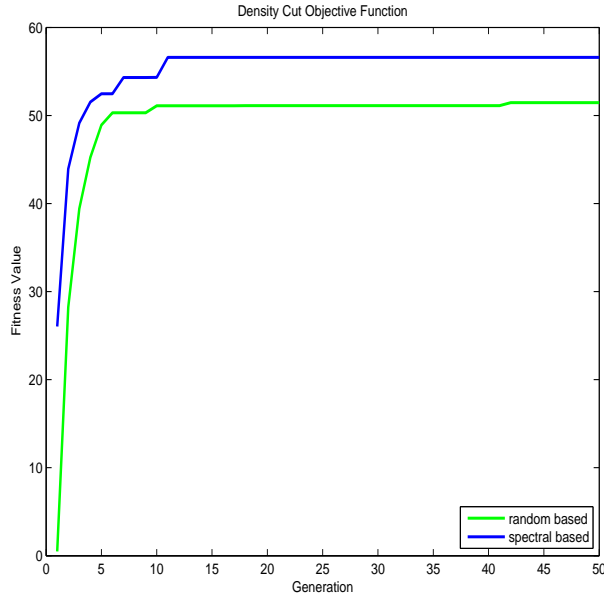
Figure 5.6 and Figure 5.7 describe the average and standard deviation of fitness values within 50 generation for arbitrarily chosen run. It is obvious that the individuals are really intending to more exploitation of the search space particularly in the case with density cut objective function.

Figure 5.8 shows the average of the best fitness values within 50 generations over 10 runs for the considered objective functions (density cut, max-min cut, normalized cut and ratio cut). As seen in this figure, the approach based on density cut fitness function has approximately the same best fitness values within 50 generations over 10 runs, i.e , the best fitness values in the  $i$ th generation are approximately similar over 10 runs, while the best fitness values computed by

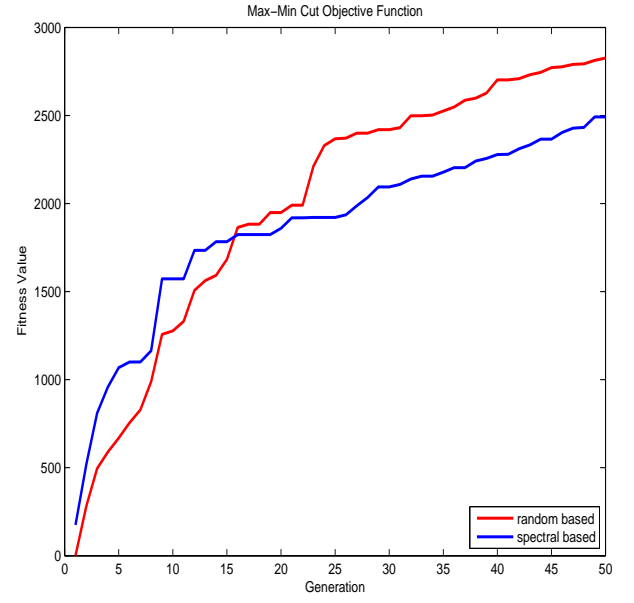


using the other objective functions (max-min cut, normalized cut and ratio cut are slightly different, (see Figure 5.8 there are a slight fluctuating).

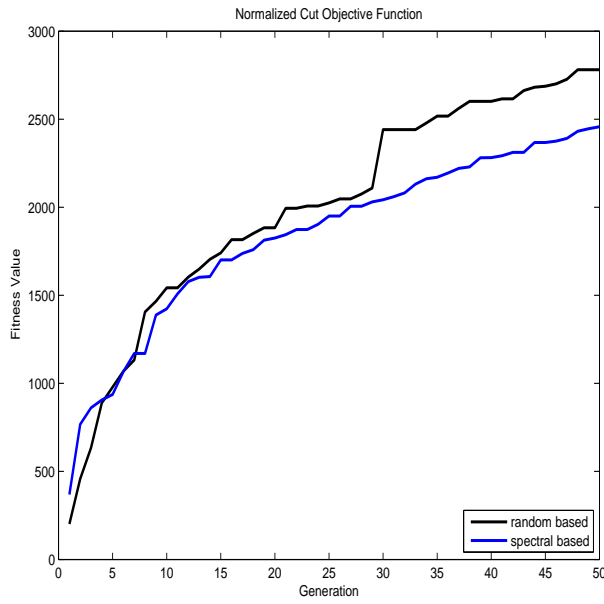
Figure 5.9 shows the standard deviation of the best fitness values within 50 generations over 10 runs for the considered objective functions (density cut, max-min cut, normalized cut and ratio cut). As seen in this figure, considering the density cut objective function, in the first 15 generations the variety of the fitness values are considerably higher (more fluctuation more variation) then the variety becomes lower until reaching to the 45th generation it becomes stable. On the other hand, using the other objective functions (max-min cut, normalized cut and ratio cut), the standard deviation does not reach to stability even after 50 generations, i.e, there exist variety among the best fitness values within 50 generations over 10 runs for each objective function.



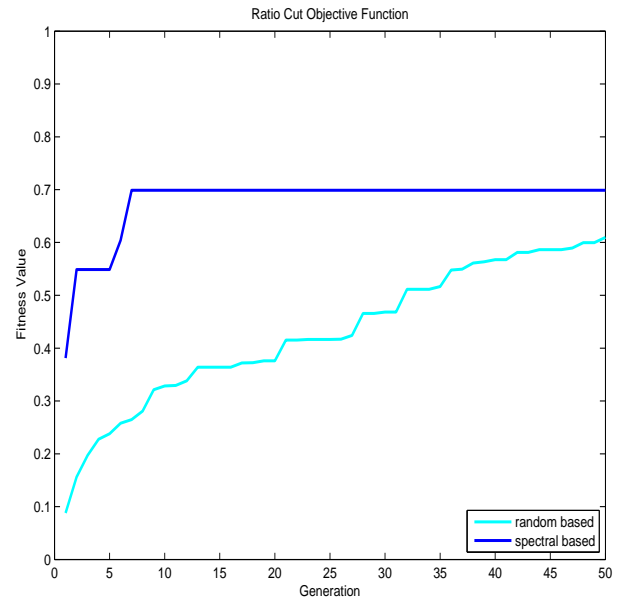
(a)



(b)

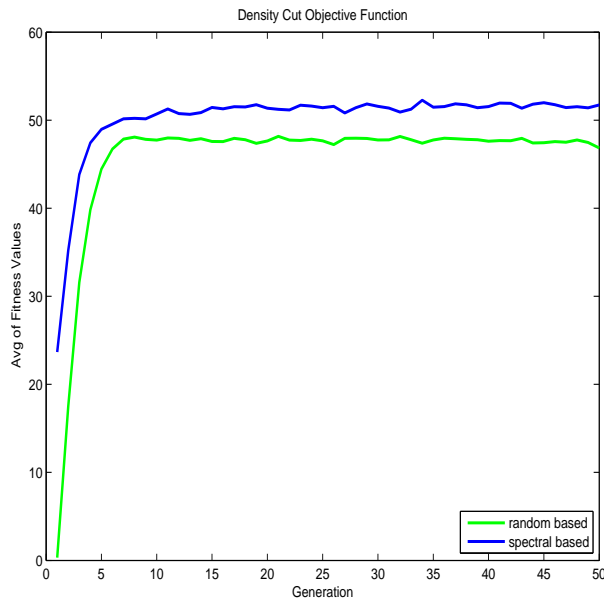


(c)

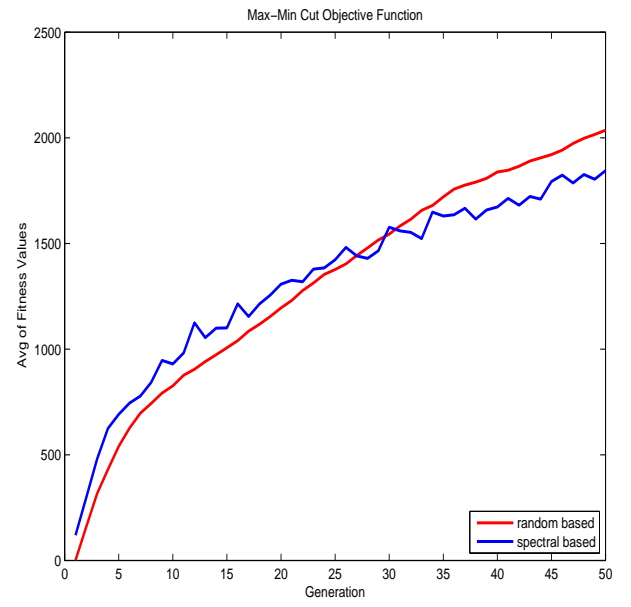


(d)

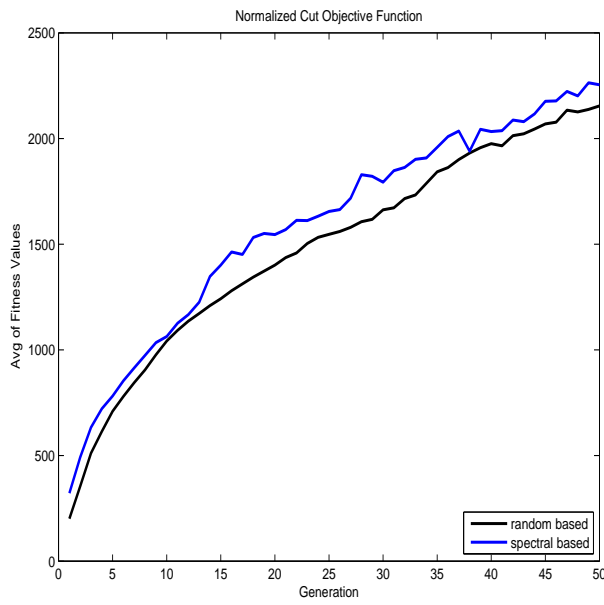
Figure 5.5: Best fitness value of four objective functions for a particular run.



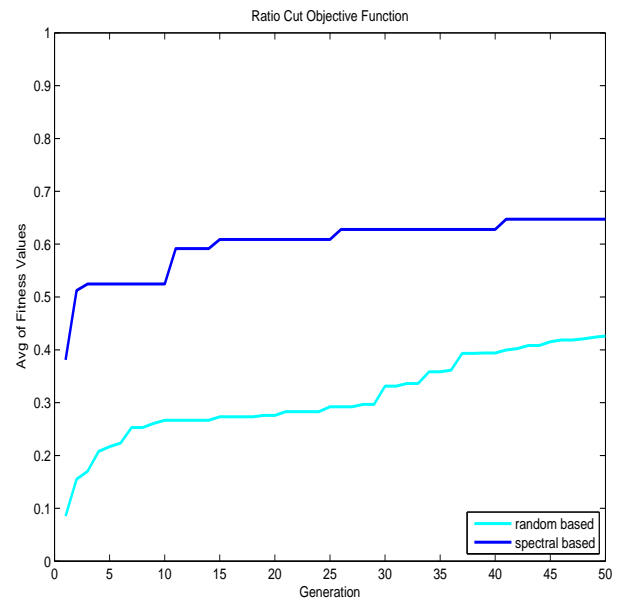
(a)



(b)

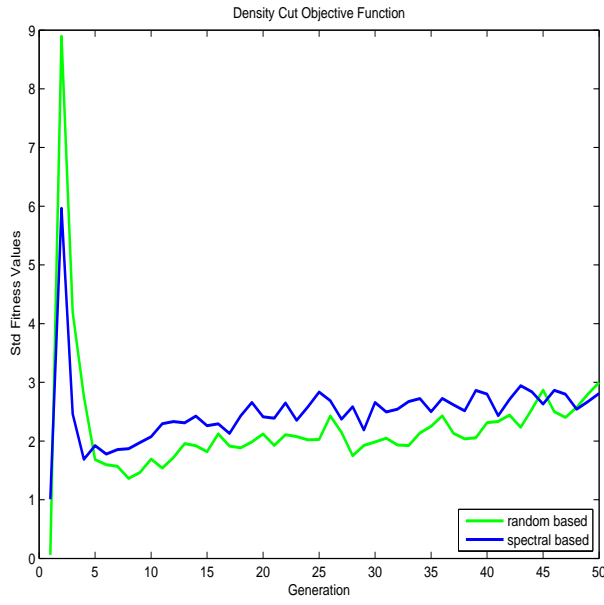


(c)

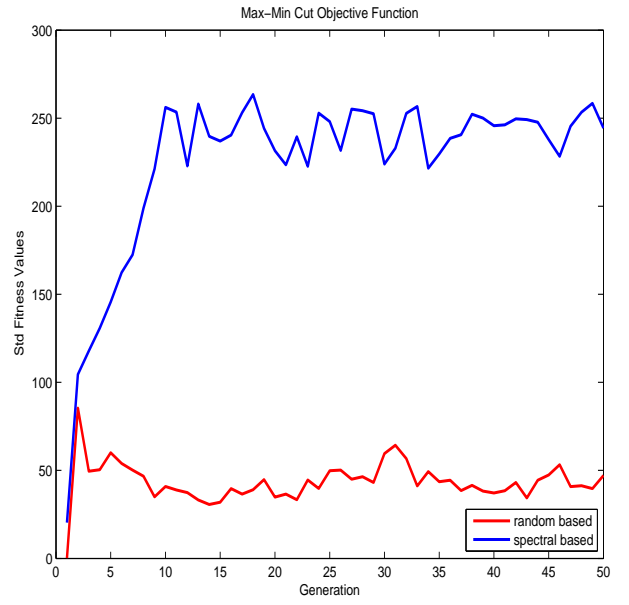


(d)

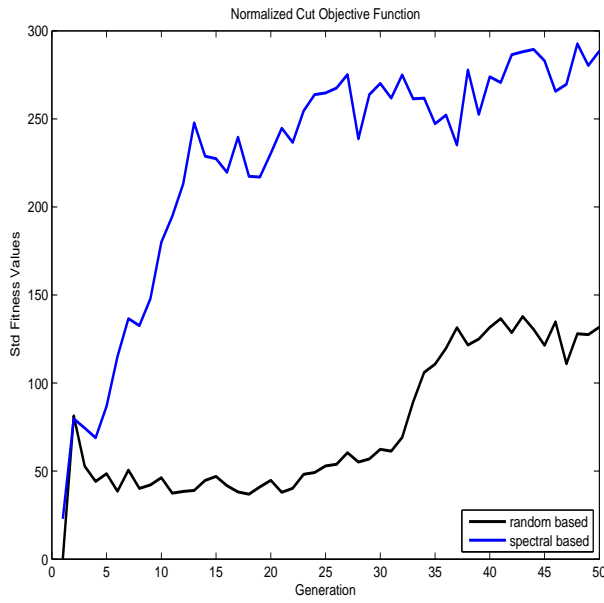
Figure 5.6: Average of fitness values of four objective functions for a particular run.



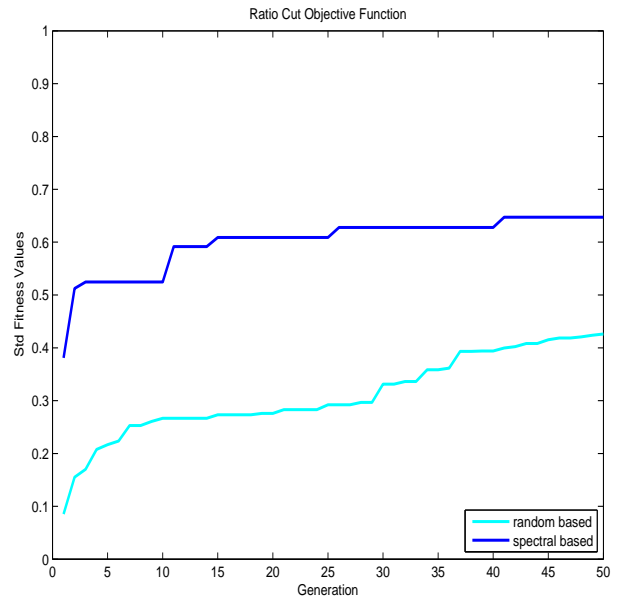
(a)



(b)

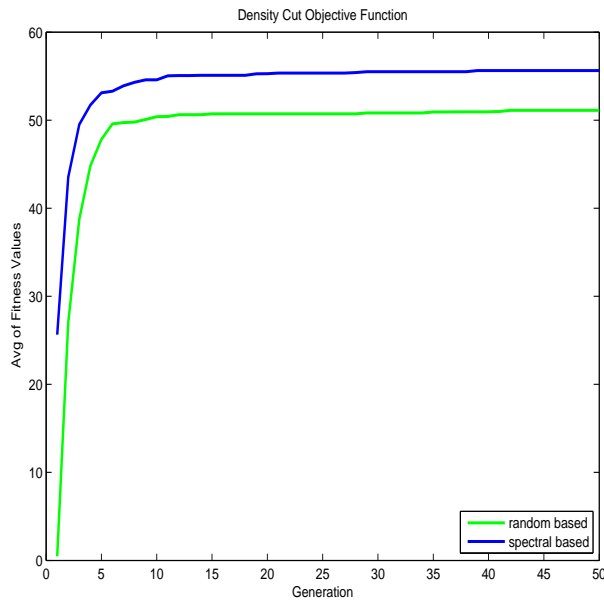


(c)

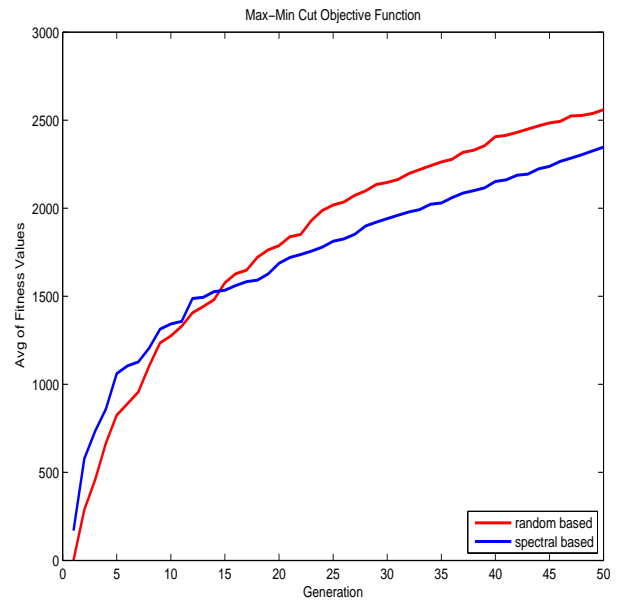


(d)

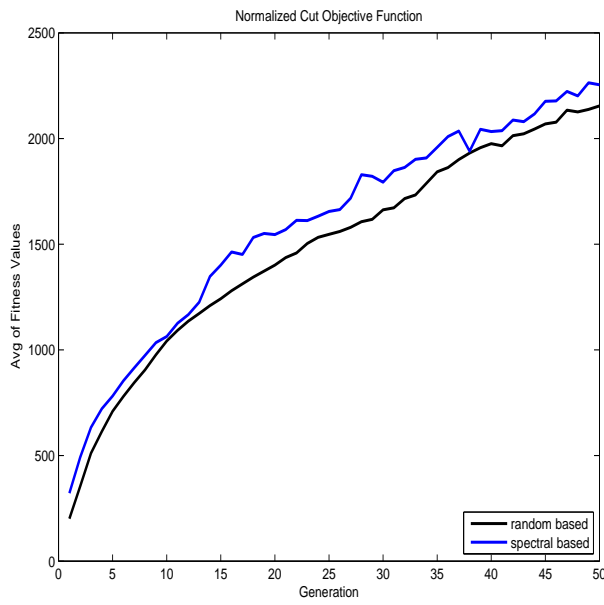
Figure 5.7: Standard deviation of fitness values of four objective functions for a particular run.



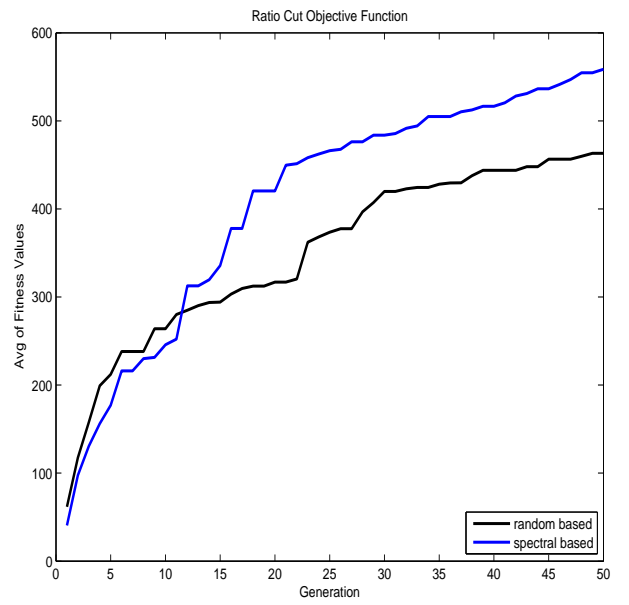
(a)



(b)

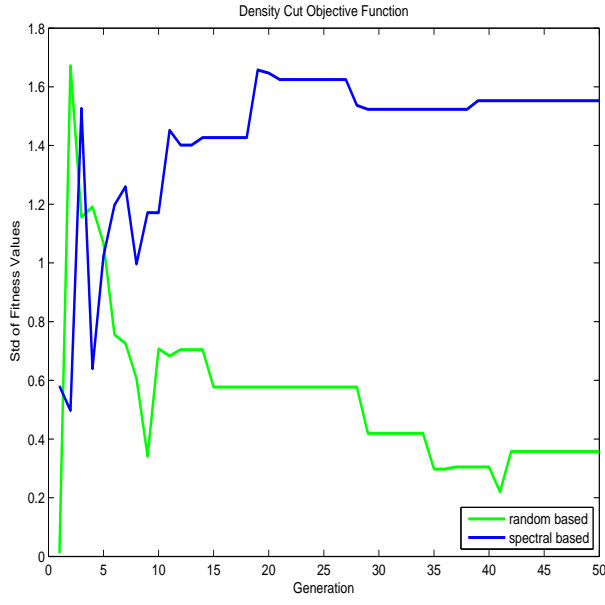


(c)

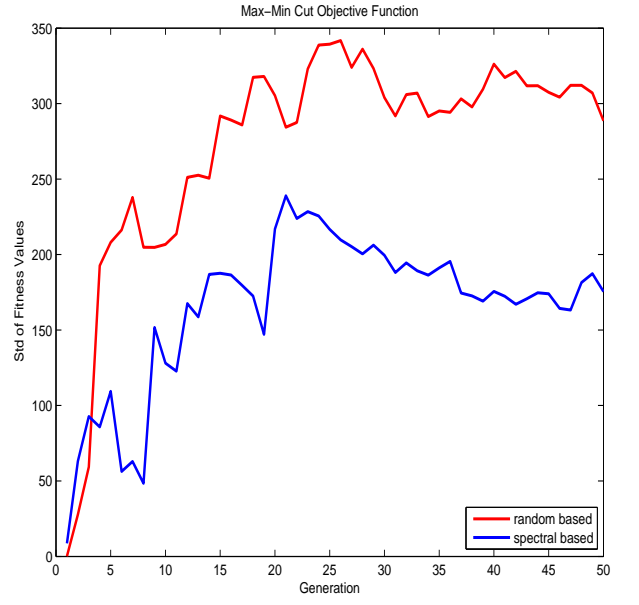


(d)

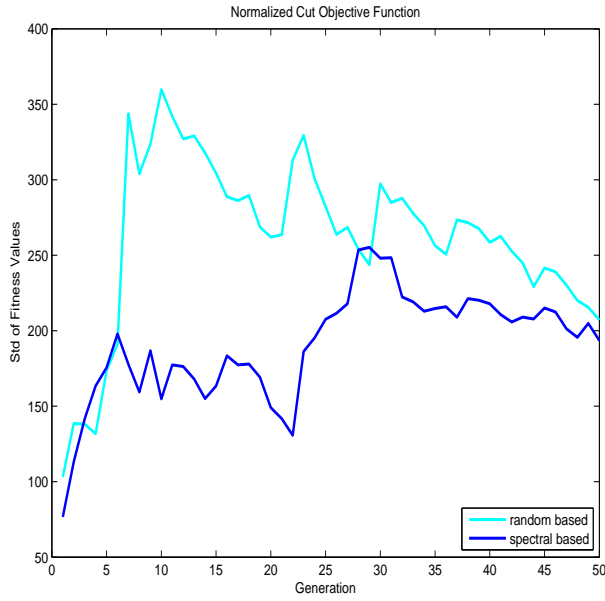
Figure 5.8: Average of the best fitness values of four objective functions over 10 runs.



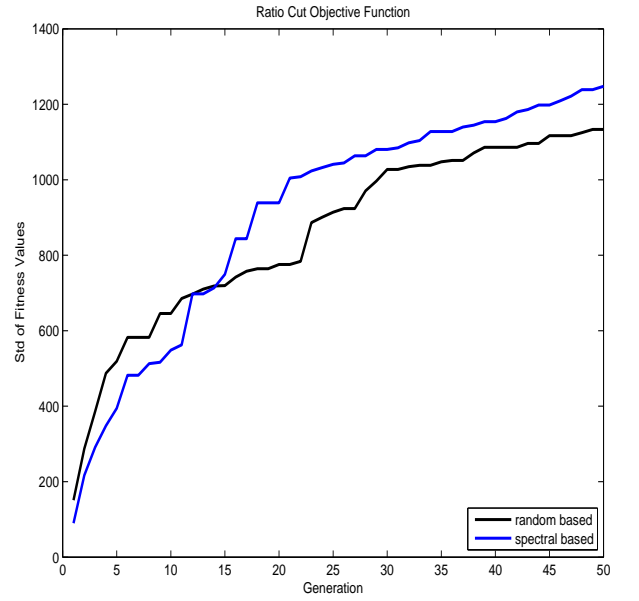
(a)



(b)



(c)



(d)

Figure 5.9: Standard deviation of the best fitness values of four objective functions over 10 runs.

### 5.2.3 Cluster validation based on functional homogeneity

Indeed, the available reference protein complexes are still uncompleted. In our study, we utilized the cellular component terms from the Gene Ontology (GO) to evaluate the quality of the identified complexes based on the fact that a group of proteins that exert their biological functions in the same cellular component can form a protein complex. We found that our method identifies several significant complexes in the Collins network. We give a snapshot of those complexes resulting from our method which based on the clusters resulting from spectral algorithm to create initial population and using density cut objective function (with size  $\leq 3$  & p-value cutoff is  $10^{-4}$ ) in Table 5.5. We use GO term finder [42] to get the most significant GO-terms, GO-id and P-values for a list of genes (predicted complex). Here,  $p$ -value is used to determine whether a specified group of genes is annotated by any GO terms at a frequency greater than that would be expected by chance. Lower  $p$ -value indicates biological significant cluster.  $p$ -value is calculated according to the following hypergeometric distribution:

$$p - value = 1 - \sum_{i=0}^{k-1} \frac{\binom{|M|}{i} \binom{|N| - |M|}{|C| - i}}{\binom{|N|}{|C|}} \quad (5.4)$$

Where  $N$  is the total number of genes,  $M$  is a list of genes that marked to the term of interest;  $C$  is the the predicted cluster and  $k$  is the number of genes that

are components of  $C$  and  $M$ .

Table 5.5: A few of the clusters in Collins network with the lowest  $p$ -values with GO components.

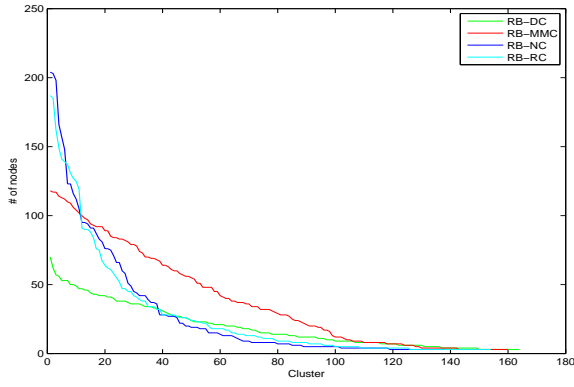
No.	Size	GO-ID	GO-Term	P-value	Num-Annotated
1	17	GO:0030880	RNA polymerase complex	3.30986E-39	100.0%
2	8	GO:0044428	nuclear part	3.70274E-05	100.0%
3	7	GO:0030126	COPI vesicle coat	1.37069E-21	100.0%
4	14	GO:0044428	nuclear part	7.2315E-10	100.0%
5	27	GO:0005739	mitochondrion	9.82318E-22	100.0%
6	29	GO:0044424	intracellular part	0.007226397	96.6%
7	18	GO:0000502	proteasome complex (sensu Eukaryota)	1.76807E-40	100.0%
8	12	GO:0005634	nucleus	3.9035E-06	100.0%
9	7	GO:0030008	TRAPP complex	1.02802E-20	100.0%
11	21	GO:0005634	nucleus	2.04087E-10	100.0%
12	10	GO:0044425	membrane part	4.18992E-10	100.0%
13	5	GO:0035097	histone methyltransferase complex	1.31389E-11	100.0%
14	5	GO:0030126	COPI vesicle coat	1.18247E-14	100.0%
15	9	GO:0016585	chromatin remodeling complex	2.37606E-17	100.0%
16	15	GO:0000502	proteasome complex (sensu Eukaryota)	2.20275E-33	100.0%
17	13	GO:0043189	H4/H2A histone acetyltransferase complex	1.21627E-39	100.0%
20	12	GO:0016514	SWI/SNF complex	4.9815E-37	100.0%
21	60	GO:0005634	nucleus	2.15384E-32	100.0%
22	81	GO:0043227	membrane-bound organelle	4.87516E-23	100.0%
23	4	GO:0031011	INO80 complex	4.13601E-07	75.0%
24	63	GO:0044464	cell part	3.42642E-05	98.4%
25	9	GO:0044445	cytosolic part	2.39611E-05	55.6%



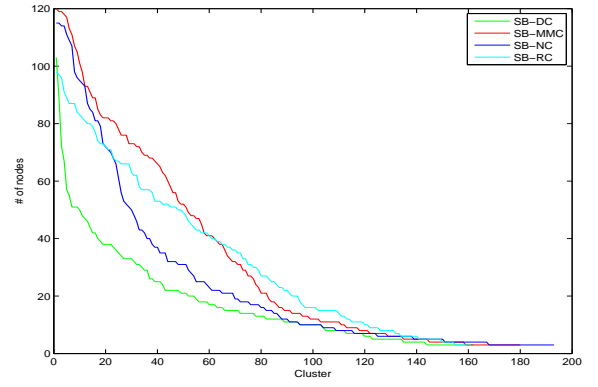
In the context of clustering PPI network, the lack of essential priori knowledge about cluster size is one of the key challenges for demonstrating the effectiveness of the developed clustering algorithm. As shown in Figure 5.10, we can address that most of the clusters resulted from MCL, MCODE, ClsusterOne and our algorithm based on the density cut objective function identify smaller-sized compared with the clusters predicted from our algorithm based on the other objective functions (ratio cut, max-min cut and normalized cut).

Figure 5.11 provides the density distribution of the clusters predicted from all considered approaches. Although the density of each cluster resulting from the density-based clustering method such as MCODE is very high, such methods discard numerous number of nodes and lose a lot of information in the considered PPI network. In general, we observe that our approach which based on density cut objective function outperforms all the others and obtains more than 100 clusters with density  $\geq 0.50$ , that is, our method can more precisely identify the modular structure in PPI network.

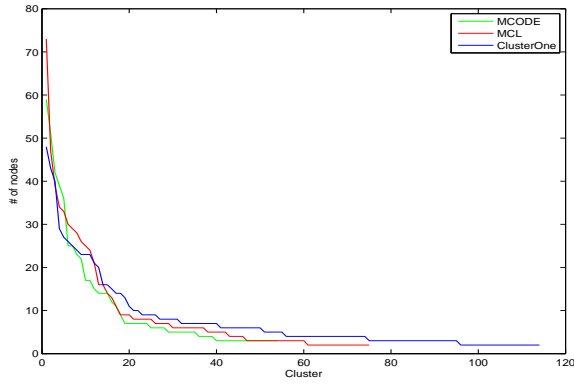
As stated in Chapter 1, each distinct biological function in the cell is carried out by a group of proteins (functional modules). Furthermore, there are some proteins be involved in multi-functional modules. Some of the clustering methods considered in this work can identify such overlapping functional modules. Figure 5.12 shows the overlapping among predicted clusters for each clustering method.



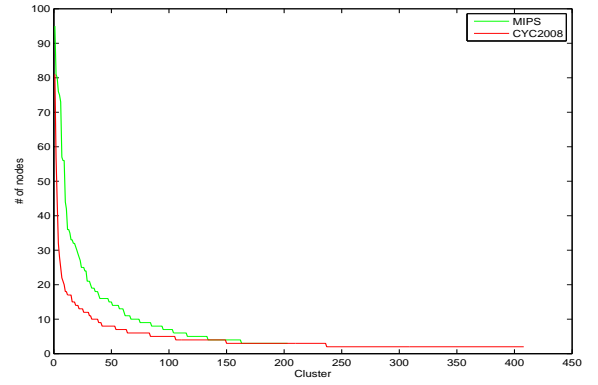
(a) Initialize population randomly.



(b) Initialize population based on spectral algorithm.

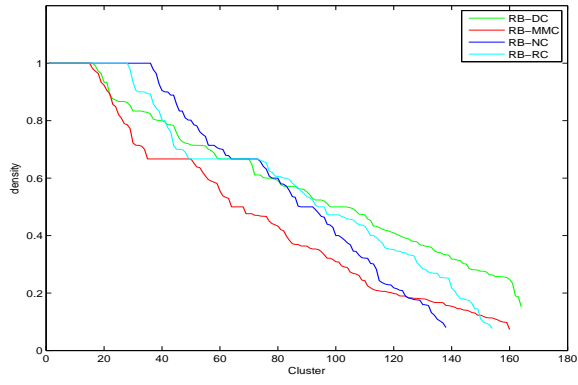


(c) Three competing clustering methods.

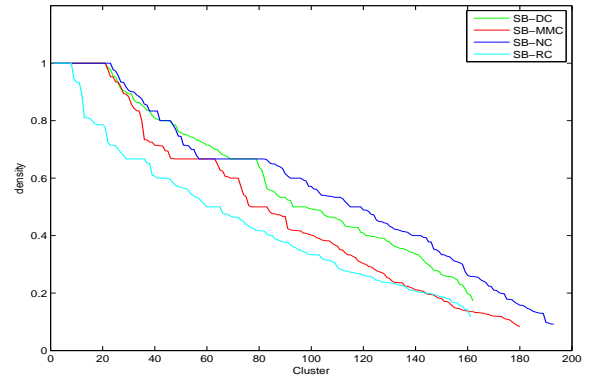


(d) Reference sets

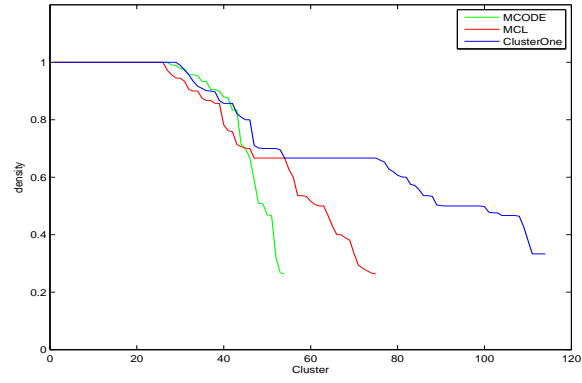
Figure 5.10: Clusters size distribution.



(a) Initialize population randomly.

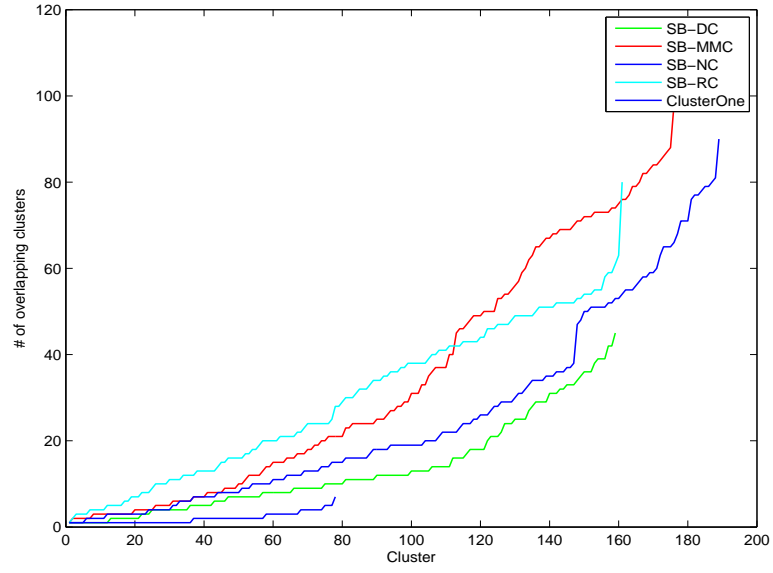


(b) Initialize population based on spectral algorithm.

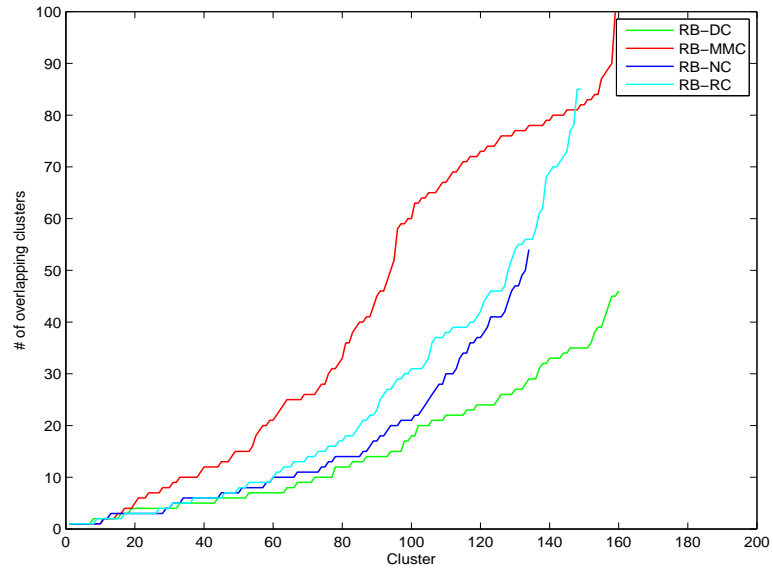


(c) Three competing clustering methods.

Figure 5.11: Clusters density distribution.



(a) Initialize population based on spectral algorithm.



(b) Initialize population randomly.

Figure 5.12: Clusters degree distribution.

# CHAPTER 6

## CYTOSCAPE PLUGIN

### (BIOCM)

The easy access to our clustering method for scientific communities is one of our goals. Thus, we have developed a user friendly Cytoscape plugin that packages all the developed algorithms required to analyze a PPI network and detect the community structure of that network. In the following sections, we provide an overview and instructions that must be followed in order to use our Bioinspired Clustering Method (BioCM) plugin.

## 6.1 Installation

To use the BioCM plugin, you must first get and download Cytoscape platform from the link bellow:

<http://www.cytoscape.org/>

Cytoscape is an open source platform used to integrate, analyze and visualize

different complex networks. It provides more than 172 plugin which developed by the community[43]. After downloading, installing and verifying that Cytoscape works correctly, you can install the BioCM plugin as follows:

1. Go to Apps → App Manager, click on this, App Manager window will pop up as shown in Figure 6.1.

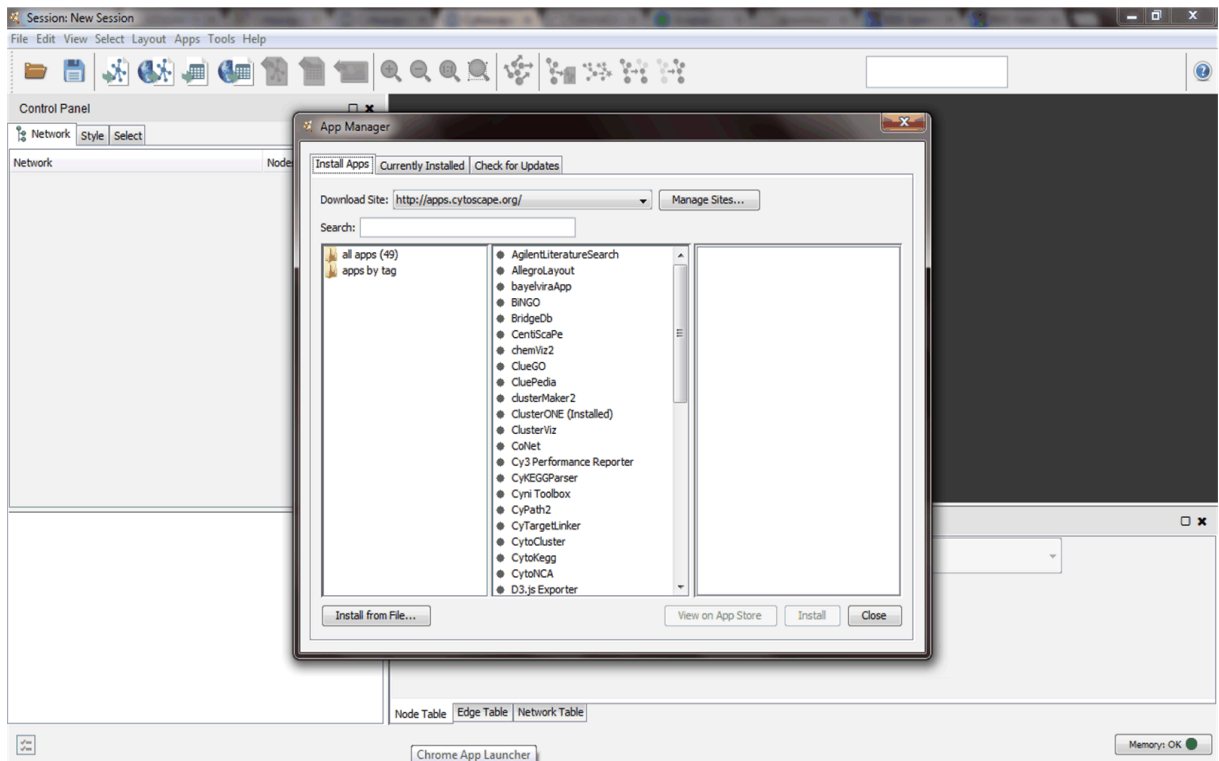


Figure 6.1: Plugin manager in Cytoscape.

2. Click the button at the left bottom of the App Manager window and select the BioCM jar file as illustrated in Figure 6.2.

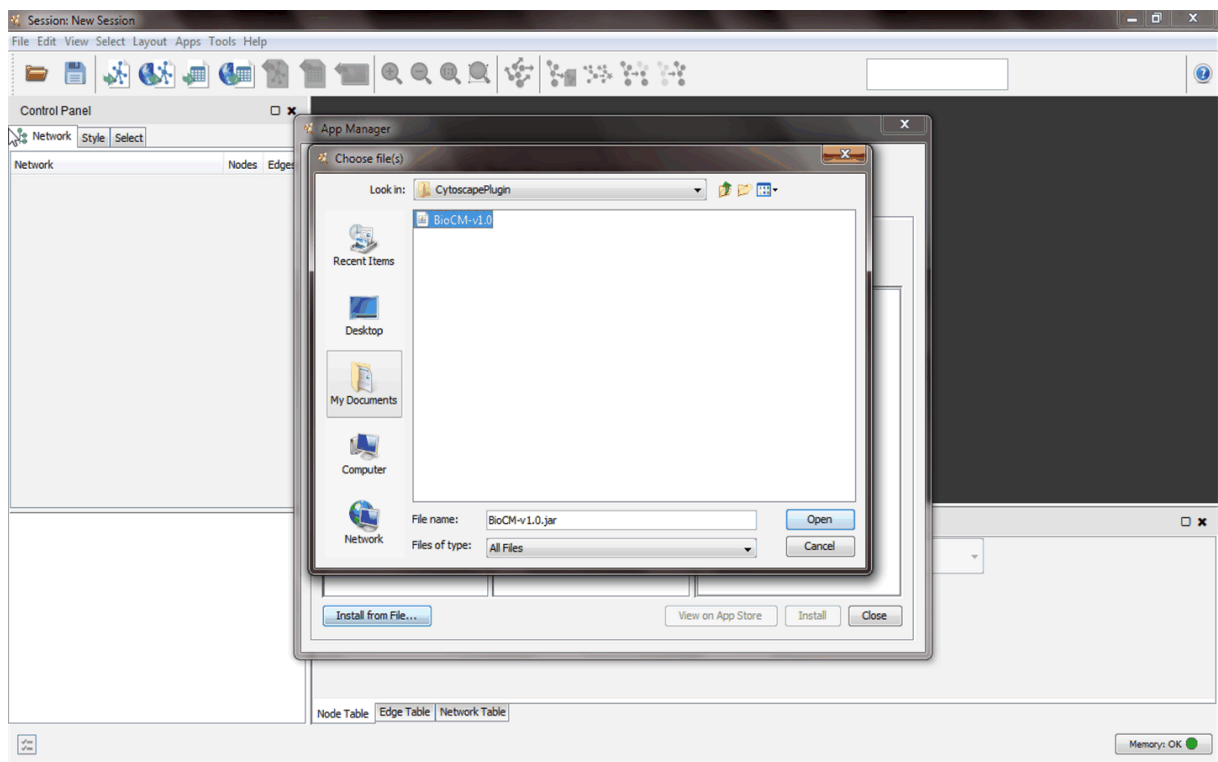


Figure 6.2: The installation of BioCM plugin.

Once you have installed BioCM on Cytoscape, make sure of two things: (i) sub-menu named (BioCM) is added to the menu (Apps); and (ii) the panel tab named (BioCM Panel) is added to the left-hand control panel of Cytoscape. as shown in Figure 6.3.

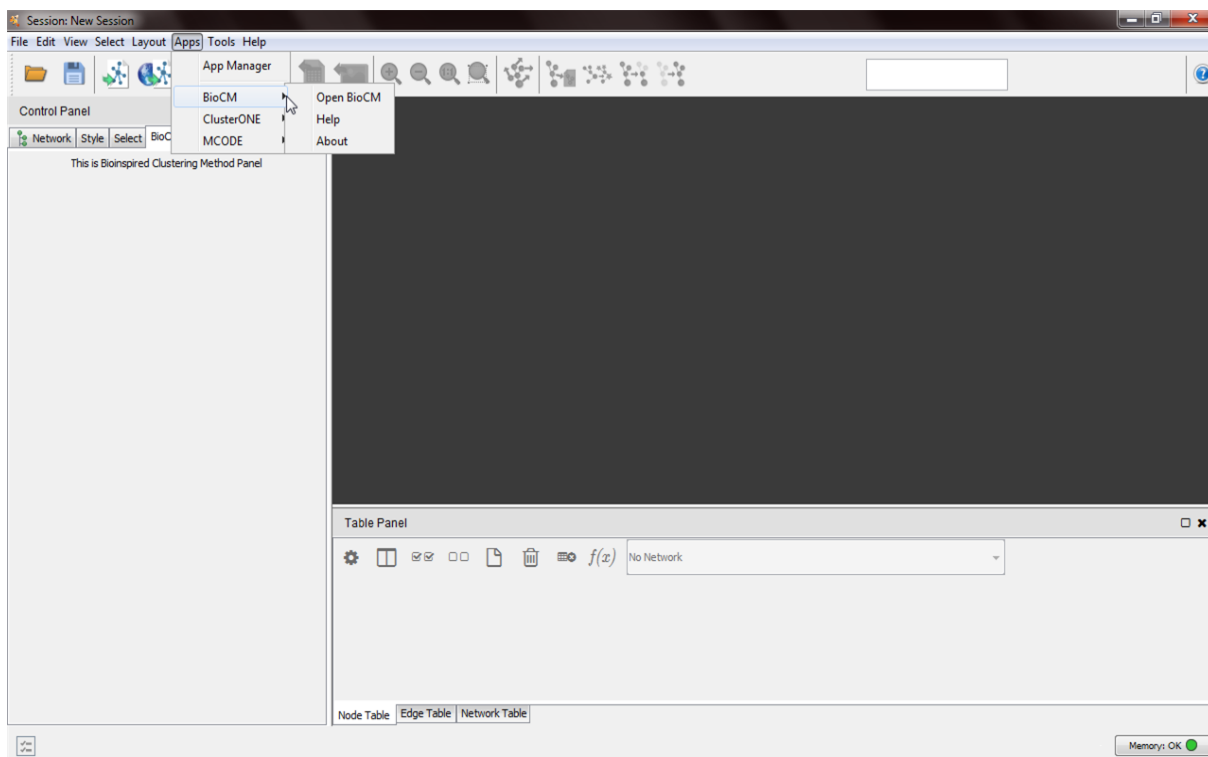


Figure 6.3: The installed BioCM plugin.



## 6.2 Running BioCM

1. The input of BioCM is the imported network file to the Cytoscape as shown in Figure 6.4.

File → import → Network → file.

Each line in the imported file specifies a source node and a destination node.

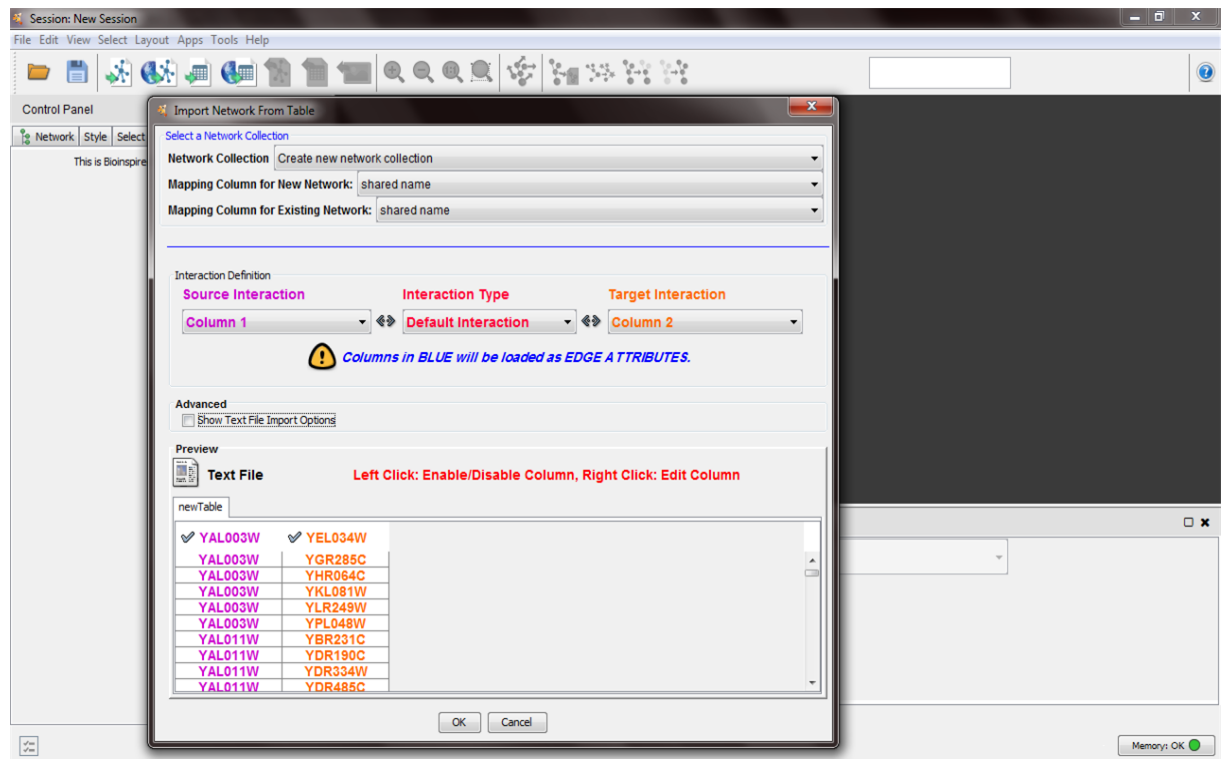


Figure 6.4: The input of BioCM plugin.

2. To analyze the imported network using our clustering method, goto → Open BioCM, then, goto the main BioCM panel appeared in the control panel of the Cytoscape shown in Figure 6.5.
3. Initialize the GA parameters and click the bottom Analyze. The result of the analyzing, as shown in Figure 6.6, is a text file including a set of overlapping clusters. Each line specifies a protein complex.

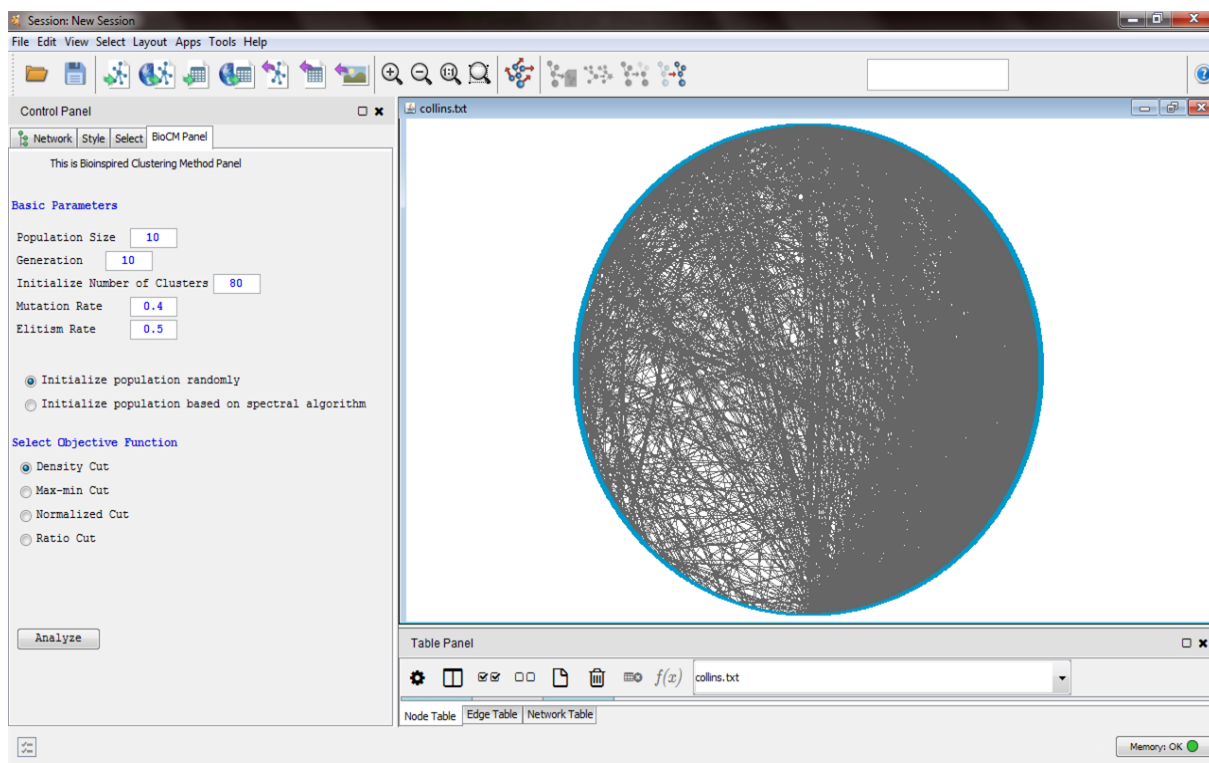


Figure 6.5: Running BioCM plugin.

```

YCR077C YDL160C YDR378C YDR473C YER112W YGL173C YGR091W YHR165C YJL124C YKL173W YLR147C YLR275W YLR438C-A YMR268C YNL147W YOR308C YPR082C YPR178W
YBR279W YER125W YER164W YGL019W YGL207W YGL244W YGR090W YHR131C YIL035C YKL088W YLR407W YLR418C YML069W YMR172W YOL145C YOR039W YOR061W YOR123C
YAL032C YBR065C YDL209C YDR364C YDR416W YER165W YGL128C YGR013W YGR278W YKR022C YLL036C YLR117C YMR125W YMR213W YOR159C YPL151C YPR182W
YAL026C YBR025C YBR127C YDL185W YEL051W YFR009W YGR020C YHR039C-A YJR033C YJR121W YKL080W YKL119C YLR447C YOR270C YOR332W YPR036W YPR149W
YBR279W YER164W YGL207W YGL244W YGR090W YIL035C YIL084C YKL088W YKR072C YLR418C YML016C YML069W YOL054W YOL145C YOR039W YOR054C YOR061W
YCR035C YDL111C YDR280W YGR095C YGR158C YGR195W YHR069C YHR081W YNL189W YNL232W YNR024W YOL021C YOL142W YOR001W YOR076C YPR189W
YBL026W YBR055C YBR152W YDL098C YDR473C YER029C YER172C YGR075C YGR091W YGR278W YHR165C YIR009W YJL124C YKL173W YOR308C YPR178W
YBL002W YBL003C YBR009C YBR010W YBR245C YDR190C YDR224C YFR037C YGL241W YKR001C YKR048C YMR033W YMR072W YOL012C YOR058C YPR052C
YBR142W YBR189W YGL123W YHL015W YJR145C YLR340W YLR432W YML063W YMR290C YNL061W YNL178W YNL301C YOL041C YOL120C YPL198W
YAR073W YBR189W YDR450W YJL191W YJR145C YLR340W YML024W YML063W YMR290C YNL178W YNL301C YOL120C YPL012W YPL198W YPR102C
YBR189W YDR450W YGL123W YJR145C YLR197W YLR340W YML063W YMR290C YNL178W YNL301C YOL041C YOL120C YOR206W YPL131W YPL198W
YAL011W YBR231C YDL070W YDR190C YDR334W YDR485C YFL039C YGR002C YJL081C YLR085C YLR385C YLR399C YML041C YNL107W YPL235W
YBL099W YBR072W YBR127C YDL185W YDR202C YEL051W YGR020C YHR039C-A YJR033C YJR121W YKL080W YLR447C YOR270C YOR332W YPR036W
YBL099W YBR025C YBR127C YDL185W YEL051W YFR009W YGR020C YHR039C-A YJR033C YJR121W YKL080W YLR447C YOR270C YOR332W YPR036W
YOL041W YER012W YER094C YFL007W YFR050C YGL011C YGR135W YGR253C YJL001W YML092C YMR314W YOL038W YOR362C YPR103W
YAL026C YBR127C YDL185W YDR202C YEL051W YGR020C YHR039C-A YJR033C YKL080W YKL119C YLR447C YOR270C YOR332W YPR036W
YBR081C YBR198C YDR145W YDR176W YGL112C YGR252W YGR274C YLR055C YML015C YMR005W YMR227C YMR236W YPL011C YPL254W
YBL046W YDR075W YDR404C YGL070C YGR005C YGR063C YGR186W YIL021W YJL140W YML010W YOL005C YOR151C YOR224C YPR187W
YCL011C YCL037C YDL051W YER165W YGL049C YGL173C YGR162W YHL034C YJL138C YJL190C YKR059W YMR125W YOL139C YOR204W
YAL003W YDL111C YDR064W YGL213C YGR158C YGR285C YHR010W YJL080C YJR123W YKL023W YLR398C YMR116C YOR063W YPR189W
YBR084W YBR189W YHL015W YJR145C YKR081C YLR340W YML063W YMR290C YNL061W YNL178W YNL301C YPL198W YPL211W
YDR359C YEL018W YFL024C YFL039C YGR002C YHR090C YHR099W YJL081C YJR082C YNL107W YNL136W YOR244W YPR023C
YAL043C YDR195W YDR301W YGR156W YJR093C YKL018W YKL059C YLR115W YLR277C YNL222W YNL317W YOR179C YPR107C
YBR154C YDL150W YDR045C YJL011C YKL144C YKR025W YNL113W YNL151C YNR003C YOR116C YOR207C YOR224C YPR190C
YBL074C YER029C YER172C YFL017W-A YGR074W YHR156C YHR165C YKL173W YLR147C YLR275W YOR159C YPR182W
YEL018W YFL024C YFL039C YGR002C YHR090C YHR099W YJL081C YJR082C YNL107W YNL136W YOR244W YPR023C
YBL071W-A YDR385W YGR192C YIL103W YJL138C YKL060C YKL152C YKL191W YLR249W YMR083W YOL086C YOR133W
YBR154C YDR156W YJL148W YJR063W YNL113W YNL248C YOR210W YOR340C YOR341W YPR010C YPR110C YPR187W
YAL043C YDR195W YDR301W YER133W YJR093C YKL018W YKL059C YLR115W YLR277C YMR242C YOR179C YPR107C
YCR014C YDL007W YDL147W YDR108W YDR394W YDR427W YFR004W YFR052W YHR027C YKL145W YLR421C YOR259C
YDR364C YDR416W YER172C YGR278W YHR165C YJR050W YLL036C YLR117C YML049C YMR213W YPL151C YPL213W
YAL013W YBR095C YDL076C YDR207C YIL084C YMR263W YNL097C YNL330C YOL004W YPL139C YPL181W
YBR289W YDR073W YFL049W YGR275W YHL025W YJL176C YMR033W YNR023W YOR290C YPL016W YPR034W
YDL002C YDR190C YFL013C YFL039C YGL150C YJL081C YLR052W YNL059C YOR141C YPL129W YPL235W

```

Figure 6.6: SnapShot of the output of BioCM plugin.

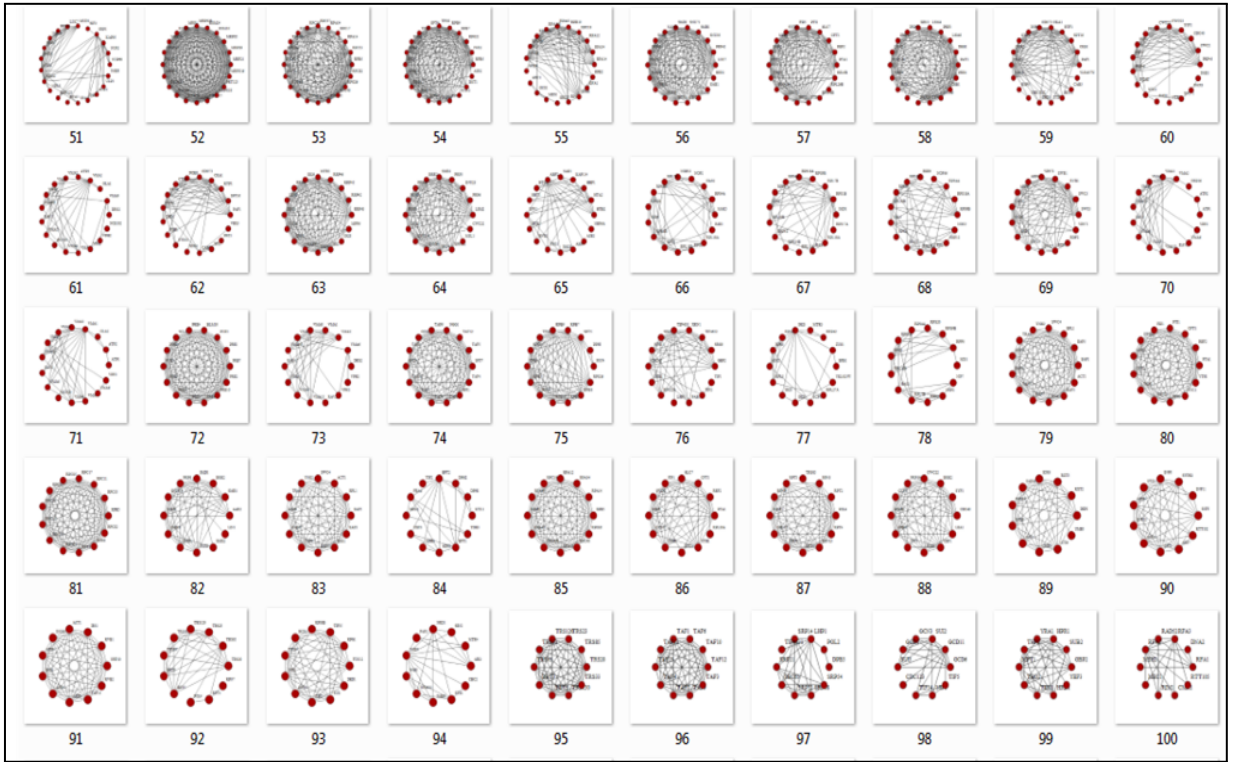


Figure 6.7: SnapShot of a visualization of the predicted clusters [using Matlab and Mathematica functions].

## CHAPTER 7

# CONCLUSION AND FUTURE WORK

We developed an approach for identifying protein complexes (i.e., clusters) in PPI networks using genetic algorithm technique. Our approach is capable of detecting densely and sparsely overlapping clusters.

We designed an objective function to allow, in overall, maximizing intra-cluster cohesion and minimizing inter-cluster coupling. Experimental results have shown that our objective function performs better than other objective functions proposed in the literature to partitioning the networks. In general, our clustering approach is more effective than existing methods (i.e., MCL, ClusterOne, and MCODE) when compared against two reference sets: MIPS and CYC2008 using three validation measures: recall, precision and f-measure. Our approach also outperformed competing approaches and is capable of effectively detecting both dense and sparsely connected biologically relevant functional modules with fewer

discards.

Future work will consider other databases and networks from other organisms, including human. Future work will also consider artificial intelligence techniques other than genetic algorithms (e.g., Swarm Intelligence) and assess performance.

# REFERENCES

- [1] K. W. Eliceiri, “Molecular expressions: Exploring the world of optics and microscopy [http: microscopy. fsu. edu](http://microscopy.fsu.edu),” *Biology of the Cell*, vol. 96, no. 6, pp. 403–405, 2004.
- [2] M. J. Zvelebil and J. O. Baum, *Understanding bioinformatics*. Garland Science, 2008.
- [3] S. Asur, D. Ucar, and S. Parthasarathy, “An ensemble framework for clustering protein–protein interaction networks,” *Bioinformatics*, vol. 23, no. 13, pp. i29–i40, 2007.
- [4] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, “A comprehensive two-hybrid analysis to explore the yeast protein interactome,” *Proceedings of the National Academy of Sciences*, vol. 98, no. 8, pp. 4569–4574, 2001.
- [5] Y. Ho, A. Gruhler, A. Heilbut, G. D. Bader, L. Moore, S.-L. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier *et al.*, “Systematic identification of

- protein complexes in *saccharomyces cerevisiae* by mass spectrometry,” *Nature*, vol. 415, no. 6868, pp. 180–183, 2002.
- [6] A.-L. Barabási, Z. N. Oltvai, and S. Wuchty, “Characteristics of biological networks,” in *Complex networks*. Springer, 2004, pp. 443–457.
- [7] A.-L. Barabasi and Z. N. Oltvai, “Network biology: understanding the cell’s functional organization,” *Nature Reviews Genetics*, vol. 5, no. 2, pp. 101–113, 2004.
- [8] B. Schwikowski, P. Uetz, and S. Fields, “A network of protein–protein interactions in yeast,” *Nature biotechnology*, vol. 18, no. 12, pp. 1257–1261, 2000.
- [9] S. V. Dongen, “Graph clustering by flow simulation,” Ph.D. dissertation, University of Utrecht, 2000.
- [10] P. Jiang and M. Singh, “Spici: a fast clustering algorithm for large biological networks,” *Bioinformatics*, vol. 26, no. 8, pp. 1105–1111, 2010.
- [11] GBaderandHogue, “An automated method for finding molecular complexes in large protein interaction networks,” *BMC Bioinformatics*, vol. 4, no. 2, p. 27 pp., 2003.
- [12] T. Nepusz, H. Yu, and A. Paccanaro, “Detecting overlapping protein complexes in protein-protein interaction networks,” *Nature Methods*, vol. 9, no. 5, pp. 471–472, 2012.



- [13] B. Adamcsek, G. Palla, I. Farkas, I. Derenyi, and T. Vicsek, “Cfinder: locating cliques and overlapping modules in biological networks,” *Bioinformatics*, vol. 22, pp. 1021–1023, 2006.
- [14] E. Becker, B. Robisson, C. E. Chapple, A. Guénoche, and C. Brun, “Multi-functional proteins revealed by overlapping clustering in protein interaction network,” *Bioinformatics*, vol. 28, no. 1, pp. 84–90, 2012.
- [15] Y. Wang and L. Gao, “Detecting protein complexes by an improved affinity propagation algorithm in protein-protein interaction networks.” *Journal of Computers*, vol. 7, no. 7, 2012.
- [16] X.-F. Zhang, D.-Q. Dai, and X.-X. Li, “Protein complexes discovery based on protein-protein interaction data via a regularized sparse generative network model,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 9, no. 3, pp. 857–870, 2012.
- [17] X.-F. Zhang, D.-Q. Dai, L. Ou-Yang, and M.-Y. Wu, “Exploring overlapping functional units with various structure in protein interaction networks,” *PloS one*, vol. 7, no. 8, p. e43092, 2012.
- [18] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in neural information processing systems*, 2000, pp. 556–562.

- [19] A. Mukhopadhyay, S. Ray, and M. De, “Detecting protein complexes in a ppi network: a gene ontology based multi-objective evolutionary approach,” *Molecular BioSystems*, vol. 8, no. 11, pp. 3036–3048, 2012.
- [20] J. H. Holland, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
- [21] L. Davis, “Handbook of genetic algorithms,” 1991.
- [22] D. E. Goldberg and J. H. Holland, “Genetic algorithms and machine learning,” *Machine learning*, vol. 3, no. 2, pp. 95–99, 1988.
- [23] D. E. Goldberg *et al.*, *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley Reading Menlo Park, 1989, vol. 412.
- [24] U. Von Luxburg, “A tutorial on spectral clustering,” *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.
- [25] C. Liu, J. Li, and Y. Zhao, “Exploring hierarchical and overlapping modular structure in the yeast protein interaction network,” *BMC genomics*, vol. 11, no. Suppl 4, p. S17, 2010.
- [26] K. Rhrissorrakrai and K. C. Gunsalus, “Mine: module identification in networks,” *BMC bioinformatics*, vol. 12, no. 1, p. 192, 2011.
- [27] Y.-R. Cho, W. Hwang, and A. Zhang, “Identification of overlapping functional modules in protein interaction networks: information flow-based ap-

- proach,” in *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*. IEEE, 2006, pp. 147–152.
- [28] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.
- [29] S. Zhang, H.-W. Liu, X.-M. Ning, and X.-S. Zhang, “A graph-theoretic method for mining functional modules in large sparse protein interaction networks,” in *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*. IEEE, 2006, pp. 130–135.
- [30] P. A. A. S. M. T. Deb, K., “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, 2002.
- [31] S. Salcedo-Sanz, S. Jiménez-Fernández, L. Carro-Calvo, J. Del Ser, J. Portilla-Figueras *et al.*, “A new grouping genetic algorithm for clustering problems,” *Expert Systems with Applications*, vol. 39, no. 10, pp. 9695–9703, 2012.
- [32] M. Tasgin, A. Herdagdelen, and H. Bingol, “Community detection in complex networks using genetic algorithms,” *arXiv preprint arXiv:0711.0491*, 2007.
- [33] C. Ding *et al.*, “A MinMaxCut spectral method for data clustering and graph partitioning,” *Proceedings of IEEE International Conference on Data Mining (ICDM)*, pp. 107–114, 2001.

- [34] C. Pizzuti, “Ga-net: A genetic algorithm for community detection in social networks,” in *Parallel Problem Solving from Nature–PPSN X*. Springer, 2008, pp. 1081–1090.
- [35] D. Lin, “An information-theoretic definition of similarity.” in *ICML*, vol. 98, 1998, pp. 296–304.
- [36] J. J. Jiang and D. W. Conrath, “Semantic similarity based on corpus statistics and lexical taxonomy,” *arXiv preprint cmp-lg/9709008*, 1997.
- [37] D. W. Huang, B. T. Sherman, and R. A. Lempicki, “Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists,” *Nucleic acids research*, vol. 37, no. 1, pp. 1–13, 2009.
- [38] J. Koljonen and J. T. Alander, “Effects of population size and relative elitism on optimization speed and reliability of genetic algorithms,” in *Proceedings of the ninth Scandinavian conference on artificial intelligence (SCAI 2006)*. Citeseer, 2006, pp. 54–60.
- [39] S. R. Collins, P. Kemmeren, X.-C. Zhao, J. F. Greenblatt, F. Spencer, F. C. Holstege, J. S. Weissman, and N. J. Krogan, “Toward a comprehensive atlas of the physical interactome of *saccharomyces cerevisiae*,” *Molecular & Cellular Proteomics*, vol. 6, no. 3, pp. 439–450, 2007.
- [40] W. J. T. B. C. E. W. S. Pu, S., “Up-to-date catalogues of yeast protein complexes,” *Nucleic Acids Research*, vol. 37 (3), pp. 825–831, 2009.

- [41] H.-W. Mewes, C. Amid, R. Arnold, D. Frishman, U. Güldener, G. Mannhaupt, M. Münsterkötter, P. Pagel, N. Strack, V. Stümpflen *et al.*, “Mips: analysis and annotation of proteins from whole genomes,” *Nucleic acids research*, vol. 32, no. suppl 1, pp. D41–D44, 2004.
- [42] E. I. Boyle, S. Weng, J. Gollub, H. Jin, D. Botstein, J. M. Cherry, and G. Sherlock, “Go:: Termfinder open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes,” *Bioinformatics*, vol. 20, no. 18, pp. 3710–3715, 2004.
- [43] Q. Wang, B. Tang, L. Song, B. Ren, Q. Liang, F. Xie, Y. Zhuo, X. Liu, and L. Zhang, “3dscapecs: application of three dimensional, parallel, dynamic network visualization in cytoscape,” *BMC bioinformatics*, vol. 14, no. 1, p. 322, 2013.

# Vitae

- Personal Information:

Name: Ahmed Abdulglil Dael Naef

Nationality: Yemeni

Date of Birth: 01 Jan 1986

Contact Details: Department of Computer Science, Taiz University,  
Taiz, Yemen

Email: *ahmdnaef@gmail.com*

- Education:

July, 2008: B.S. Computer Science, Taiz University, Yemen

May, 2014: Master of Science in Computer Science, KFUPM, Saudi  
Arabia

- Training and Experience:

Nov 2008 - Jul 2010: Instructor in Computer Science Dept., Faculty of  
Science, Taiz University, Yemen